

File 8: Ei Compendex(R) 1970-2006/Feb W3
 (c) 2006 Elsevier Eng. Info. Inc.
 File 35: Dissertation Abs Online 1861-2006/Feb
 (c) 2006 ProQuest Info&Learning
 File 65: Inside Conferences 1993-2006/Feb W4
 (c) 2006 BLDSC all rts. reserv.
 File 2: INSPEC 1898-2006/Feb W3
 (c) 2006 Institution of Electrical Engineers
 File 94: JICST-EPlus 1985-2006/Dec W1
 (c) 2006 Japan Science and Tech Corp (JST)
 File 6: NTIS 1964-2006/Feb W1
 (c) 2006 NTIS, Intl Cpyrght All Rights Res
 File 144: Pascal 1973-2006/Feb W1
 (c) 2006 INIST/CNRS
 File 434: SciSearch(R) Cited Ref Sci 1974-1989/Dec
 (c) 1998 Inst for Sci Info
 File 34: SciSearch(R) Cited Ref Sci 1990-2006/Feb W3
 (c) 2006 Inst for Sci Info
 File 99: Wilson Appl. Sci & Tech Abs 1983-2006/Jan
 (c) 2006 The HW Wilson Co.
 File 266: FEDRIP 2005/Dec
 Comp & dist by NTIS, Intl Copyright All Rights Res
 File 95: TEME-Technology & Management 1989-2006/Feb W4
 (c) 2006 FIZ TECHNIK
 File 438: Library Lit. & Info. Science 1984-2006/Jan
 (c) 2006 The HW Wilson Co
 ?

Set	Items	Description
S1	8128648	PROGRAM? ? OR SOFTWARE OR APPLICATION? ?
S2	1269756	PANEL? ? OR PANE OR PANES OR WINDOW? ? OR SCREEN? ? OR BOX OR BOXES OR FRAME OR FRAMES
S3	67270	(SERIES OR SEQUENCE? ? OR SUCCESSIV? OR SUCCESSION OR SUCC- EEDING OR SUBSEQUENT?? OR TRANSITION? OR ORDER??? OR CONSECUT- IV? OR PROGRESSION OR CHAIN??? OR TRAIN? ? OR LINK??? OR CONN- ECT???) (5N) S2
S4	2557	(NEXT OR ENSUING OR FOLLOWING) (3W) S2
S5	92325	(INVOK??? OR INVOCATION OR CALL???) (5N) (METHOD? ? OR PROCE- DURE? ? OR FUNCTION? ?)
S6	1048083	OBJECT? ? OR OO OR OOP OR OOPL OR OOPLA
S7	10245	S1 (5N) (LEGACY OR OLD OR OLDER OR OUTDATED)
S8	347184	S1 (5N) (MODERN OR NEW OR UPDAT? OR CURRENT OR RECENT)
S9	13948	S1 AND S3:S4
S10	1504	S6 AND S9
S11	12	S5 AND S10
S12	797	S10 AND (METHOD? ? OR PROCEDURE? ? OR FUNCTION? ?)
S13	71	S10 AND S7:S8
S14	79	S11 OR S13
S15	55	RD (unique items)
S16	30	S15 NOT PY=1998:2006
S17	552	S1 (5N) (OUT(1W) DATE OR PREEXIST? OR PRE() EXIST?)
S18	74	S1 (5N) (BACKWARD() COMPAT?)
S19	2	S10 AND S17:S18
S20	12615	S2 (7N) (GRAPH??? OR DIGRAPH?)
S21	6646	S1 AND S20
S22	795	S6 AND S21
S23	41	(S7:S8 OR S17:S18) AND S22
S24	34	RD (unique items)
S25	26	S24 NOT (S16 OR PY=1998:2006)

S26 15185 S1 AND S2 AND (S7:S8 OR S17:S18)
S27 1124 S6 AND S26
S28 2212 (S7 OR S17) AND (S8 OR S18)
S29 21 S27 AND S28
S30 15 RD (unique items)
S31 8 S30 NOT PY=1998:2006
S32 26984 S1(5N)(REUSE? ? OR REUSING OR REUSABLE OR REUSABILITY OR R-
E()(USE OR USING OR USABLE OR USABILITY))
S33 31 S10 AND S32
S34 15 S22 AND 32
S35 68 S27 AND S32
S36 112 S33:S35
S37 88 RD (unique items)
S38 22 S37 NOT (S11 OR S16 OR S19 OR S25 OR S31 OR PY=1996:2006)

T/5/1,3,4,5,6,7,16,19-21,25,29,30

16/5/1 (Item 1 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04623707 E.I. No: EIP97023519654

Title: Is GUI programming a database research problem?

Author: Goyal, Nita; Hoch, Charles; Krishnamurthy, Ravi; Meckler, Brian; Suckow, Michael

Corporate Source: Hawlett Packard Labs

Conference Title: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data

Conference Location: Montreal, Can Conference Date: 19960604-19960606

Sponsor: ACM SIGMOD

E.I. Conference No.: 45963

Source: SIGMOD Record (ACM Special Interest Group on Management of Data) v 25 n 2 June 1996.. p 517-528

Publication Year: 1996

CODEN: SRECD8

Language: English

Document Type: CA; (Conference Article) Treatment: G; (General Review); T; (Theoretical)

Journal Announcement: 9704W1

Abstract: Programming nontrivial GUI **applications** is currently an arduous task. Just as the use of a declarative language simplified the programming of database **applications**, we ask whether we can do the same for GUI programming? Can we then import a large body of knowledge from database research? We answer these questions by describing our experience in building nontrivial GUI **applications** initially using C plus plus programming and subsequently using Logic plus plus, a higher order Horn clause logic language on complex **objects** with **object**-oriented features. We abstract a GUI **application** as a set of event handlers. Each event handler can be conceptualized as a **transition** from the **old screen / program** state to a **new screen/program** state. We use a data centric view of the screen/ **program** state (i.e., every entity on the screen corresponds to proxy datum in the **program**) and express each event handler as a query dependent update, albeit a complicated one. To express such complicated updates we use Logic plus plus. The proxy data are expressed as derived views that are materialized on the screen. Therefore, the system must be active in maintaining these materialized views. Consequently, each event handler is conceptually an update followed by a fixpoint computation of the proxy data. Based on our experience in building the GUI system, we observe that many database techniques such as view maintenance, active DB, concurrency control, recovery optimization as well as language concepts such as higher order logic are useful in the context of GUI programming. (Author abstract) 14 Refs.

Descriptors: *Graphical user interfaces; Computer systems programming; Query languages; **Object** oriented programming; Optimization; C (programming language); Logic programming; Computational complexity

Identifiers: Proxy data; Horn clause logic language

Classification Codes:

723.1.1 (Computer Programming Languages)

722.2 (Computer Peripheral Equipment); 723.1 (Computer Programming);

723.3 (Database Systems); 921.5 (Optimization Techniques); 721.1

(Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory)

722 (Computer Hardware); 723 (Computer Software); 921 (Applied Mathematics); 721 (Computer Circuits & Logic Elements)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

16/5/3 (Item 3 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04363119 E.I. No: EIP96033109700

Title: Data acquisition developers upgrade to Windows 95

Author: Studt, Tim

Source: Research & Development (Barrington, Illinois) v 38 n 2 Feb 1996.

p 65-68

Publication Year: 1996

CODEN: REDEEA

Language: English

Document Type: JA; (Journal Article) Treatment: G; (General Review)

Journal Announcement: 9605W3

Abstract: Within six months of Win95's release, more than 75% of data acquisition **software** developers have introduced versions of their **software** compatible with Microsoft's **new** operating system. Win95's ability to run 32-bit **applications** is one reason **software** developers have upgraded their data-acquisition **programs**. However, not all have gone to 32-bit releases due to concerns over whether Win95 will truly become the dominant system. Meanwhile, most upgrades have gone beyond mere compatibility additions in that they have also added improvements to ease of use, flexibility, and compatibility with other **software** products.

Descriptors: ***Softwar** e engineering; Computer **software**; Data acquisition; Computer systems; Graphical user interfaces; **Object** oriented programming; Computer aided **software** engineering; **Program** debugging; BASIC (programming language); C (programming language)

Identifiers: **Software** Package **Windows** 95; Data acquisition□**software**□
; **Object** **linking** and embedding; Dynamic data exchange

Classification Codes:

723.1.1 (Computer Programming Languages)

723.5 (Computer Applications); 723.1 (Computer Programming); 723.2 (Data Processing); 722.4 (Digital Computers & Systems); 722.2 (Computer Peripheral Equipment)

723 (Computer Software); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING)

16/5/4 (Item 4 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04350875 E.I. No: EIP96023043866

Title: Building virtual instruments with OLE controls

Author: Pasquarette, John

Source: EE: Evaluation Engineering v 35 n 2 Feb 1996. p 22-25

Publication Year: 1996

CODEN: EVENAE ISSN: 0149-0370

Language: English

Document Type: JA; (Journal Article) Treatment: G; (General Review)

Journal Announcement: 9604W3

Abstract: The introduction of Windows 95 marks the first step toward real code sharing and reuse between widespread **applications** through a **new** technology called **object** linking and embedding (OLE). OLE is a very broad term that describes a wide variety of operations. However, it embodies a very simple concept which is making different **software** components work together better.

Descriptors: *Computer operating systems; User interfaces; Personal

computers; Command and control systems; Computer **software** ; Data structures; C (programming language); **Object** oriented programming; **Software** engineering; Computer architecture
 Identifiers: **Software** Package **Windows** 95; Code reuse; ☐Object ☐
linking and embedding; Dynamic data exchange; Compatibility; Configuration
 ; Programmability
 Classification Codes:
 723.1.1 (Computer Programming Languages)
 723.1 (Computer Programming); 722.2 (Computer Peripheral Equipment);
 722.4 (Digital Computers & Systems); 731.1 (Control Systems); 723.2
 (Data Processing)
 723 (Computer Software); 722 (Computer Hardware); 731 (Automatic
 Control Principles)
 72 (COMPUTERS & DATA PROCESSING); 73 (CONTROL ENGINEERING)

16/5/5 (Item 5 from file: 8)
 DIALOG(R)File 8: Ei Compendex(R)
 (c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04272290 E.I. No: EIP95092858820
Title: Image sequence analysis for updating a 4-dimensional GIS of coastal dunes
 Author: Mulder, Nanno J.
 Corporate Source: Univ of Twente, Enschede, Neth
 Conference Title: Proceedings of the 1995 International Geoscience and Remote Sensing Symposium. Part 1 (of 3)
 Conference Location: Firenze, Italy Conference Date: 19950710-19950714
 Sponsor: IEEE; URSI
 E.I. Conference No.: 43564
 Source: International Geoscience and Remote Sensing Symposium (IGARSS) v 1 1995., 95CH35770. p 96-98
 Publication Year: 1995
 CODEN: IGRSE3
 Language: English
 Document Type: CA; (Conference Article) Treatment: A; (Applications); T
 ; (Theoretical)
 Journal Announcement: 9512W3
Abstract: The **application** problem is to **update** the digital elevation model of a dune area with RS data taken, with a high degree of overlap, as a video sequence from a light aeroplane. The problem in image analysis is how to estimate geometric parameters of plants and sand volumes from a sequence of images in which the same **object** is imaged some 128 times from different look-angles. As the classical methods for correspondence analysis of stereo images are not suitable for our purpose, we develop a new method based on prediction and parameter estimation. After estimating camera and platform parameters, for each image frame, we estimate the parameters of a piecewise linear **object** element for each intersect of **object** with a scanline plane. The method is least squares optimal in the parameter domain over a large set of observations. The method solves most occlusion problems. (Author abstract)
 Descriptors: *Image analysis; Image segmentation; Geographic information systems; Shore protection; Mathematical models; Parameter estimation; Forecasting; Least squares approximations; Stereo vision; Cameras
 Identifiers: Image sequence; Four-dimensional; Coastal dunes; Digital elevation model; Video **sequence** ; Image **frame** ; Platform parameters; Scanline plane
 Classification Codes:
 723.2 (Data Processing); 723.3 (Database Systems); 407.1 (Maritime Structures); 921.6 (Numerical Methods); 731.1 (Control Systems); 912.2

(Management)

723 (Computer Software); 407 (Maritime & Port Structures); 921

(Applied Mathematics); 731 (Automatic Control Principles); 912

(Industrial Engineering & Management)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS); 73

(CONTROL ENGINEERING); 91 (ENGINEERING MANAGEMENT)

16/5/6 (Item 6 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

03919242 E.I. No: EIP94081363500

Title: Framework for CS1 and CS2 laboratories

Author: Waller, William A.

Corporate Source: Univ of Houston, Houston, TX, USA

Conference Title: Proceedings of the 25th SIGCSE Technical Symposium on
Computer Science Education

Conference Location: Phoenix, AZ, USA Conference Date:
19940310-19940311

Sponsor: ACM

E.I. Conference No.: 20684

Source: SIGCSE Bulletin (Association for Computing Machinery, Special
Interest Group on Computer Science Education) v 26 n 1 Mar 1994. p 198-202

Publication Year: 1994

CODEN: SIGSD3 ISSN: 0097-8418 ISBN: 0-89791-646-8

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); X;
(Experimental)

Journal Announcement: 9409W5

Abstract: The development and implementation of closed laboratories in the undergraduate curriculum continues to be an important trend in computer science education. Most textbooks intended for use in CS1 and CS2 level courses are now supplemented with closed laboratory manuals. Many instructors have reported efforts to incorporate closed labs into other courses as well. The National Science Foundation has funded several projects of both local and national scope aimed at promoting the integration of closed laboratories into the undergraduate curriculum. In this paper we describe the experimental authorware system PHIL, which is designed to create laboratory exercises for computer science courses, particularly CS1 and CS2. The outputs of this system are interactive MS Windows 'laboratory' documents, which guide students through a series of activities and questions. These activities and questions are presented via windows which provide the students with objectives, instructions, and different types of controls for student responses (such as text edit fields). Activity **windows** can be **linked** with useful external

applications, such as compilers, through buttons to permit easy access. Menu choices allow the student to save and retrieve lab documents from disk, as well as print equivalent hard-copy versions of the documents. Authors using the system can create **new lab programs** by customizing and recombining pre-existing activities, or can create entirely new activities based on predefined templates. The PHIL system is based on parallel hierarchies of lab activity **objects** developed using Borland C plus plus and ObjectWindows, the Borland framework for creating MS Windows

applications. The goals of this system are to encourage reuse of lab material developed for CS1 and CS2 courses, to provide a convenient and extensible environment for developing CS lab material, and to provide a stimulating and sophisticated closed lab environment for lower division CS courses. (Author abstract) 6 Refs.

Descriptors: *Computer **software**; Computer science; Laboratories;

Engineering education; Computer operating systems; Interactive computer graphics; **Program** compilers; Information retrieval; Magnetic disk storage ; Hierarchical systems

Identifiers: Undergraduate curriculum; Authorware systems; Windows laboratory document; Activity window; Computer science courses; Borland framework; **Software** package - PHIL

Classification Codes:

723.1 (Computer Programming); 723.5 (Computer Applications); 901.2 (Education); 722.4 (Digital Computers & Systems); 903.3 (Information Retrieval & Use)

723 (Computer Software); 901 (Engineering Profession); 722 (Computer Hardware); 903 (Information Science)

72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING)

16/5/7 (Item 7 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

03584762 E.I. Monthly No: EI9304045779

Title: EmbeddedButtons: Supporting buttons in documents.

Author: Bier, Eric A.

Corporate Source: Xerox Palo Alto Research Cent, Palo Alto, CA, USA

Source: ACM Transactions on Information Systems v 10 n 4 Oct 1992 p 381-407

Publication Year: 1992

CODEN: ATISSET ISSN: 1046-8188

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9304

Abstract: EmbeddedButtons is a library of routines and a runtime kernel that support the integration of buttons into document media, including text and graphics. Existing document editors can be modified to participate in this open architecture with the addition of a few simple routines. Unlike many button systems that insert special button **objects** into document media, this system supports turning existing document **objects** into buttons. As a consequence, buttons inherit all of the attributes of normal document **objects** , and the appearance of buttons can be edited using operations already familiar to users. Facilities are provided for **linking** buttons to **application windows** so that documents can serve as

application control panels. Hence, user interface designers can lay out control panels using familiar document editors rather than special-purpose tools. Three classes of buttons have been implemented, including buttons that pop up a menu and buttons that store and display the value of a variable. **New** button classes, editors, and **applications** can be added at run time. Two editors, one for text and one for graphics, currently participate in the architecture. (Author abstract) 28 Refs.

Descriptors: *FILE EDITORS; INTERACTIVE COMPUTER SYSTEMS; HUMAN ENGINEERING; USER INTERFACES; COMPUTER **SOFTWARE** ; COMPUTER ARCHITECTURE; SUBROUTINES

Identifiers: EMBEDDEDBUTTONS; ACTIVE DOCUMENTS; TEXT EDITING; INTERACTION TECHNIQUES; RAPID PROTOTYPING

Classification Codes:

723 (Computer Software); 722 (Computer Hardware); 461 (Biotechnology)

72 (COMPUTERS & DATA PROCESSING); 46 (BIOENGINEERING)

16/5/16 (Item 4 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

05928804 INSPEC Abstract Number: C9505-6140D-057

Title: VIM language paradigm

Author(s): Mirenkov, N.

Author Affiliation: Aizu Univ., Fukushima, Japan

p.569-80

Editor(s): Buchberger, B.; Volkert, J.

Publisher: Springer-Verlag, Berlin, Germany

Publication Date: 1994 Country of Publication: West Germany xvi+889

PP.

Conference Title: Proceedings of International Conference on Parallel Processing: CONPAR '94 - VAPP VI

Conference Date: 6-8 Sept. 1994 Conference Location: Linz, Austria

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: A visual language paradigm **called** VIM (VISualization of **Methods**), for the interactive specification of **application** algorithms is proposed. The approach is based on a set of computational schemes ("shapes" of computation) presented by color figures, pictures and animation films with sound accompaniment. Each film is related to a **series** of **frames** (computational steps), and it reflects some knowledge about the data processing. Each frame "brightens" up a substructure of the data for which operations should be specified. As a rule, this substructure is a set of points and/or moving **objects** in a multi-dimensional space-time. A user embeds his algorithm into computational schemes by making these schemes more precise. In fact, he defines the specification by creating a **new** film. The corresponding **program** (sequential or parallel) is then generated automatically. (10 Refs)

Subfile: C

Descriptors: automatic programming; colour graphics; computer animation; data visualisation; interactive programming; spatial data structures; specification languages; visual languages

Identifiers: VIM language paradigm; visual language; methods visualization; interactive specification; **application** algorithms; computational schemes; computation shapes; color figures; pictures; animation films; sound accompaniment; data processing; data substructure brightening; operations specification; moving **objects** ; multi-dimensional space-time; frames; computational steps; sequential **programs** ; parallel **programs** ; automatic **program** generation

Class Codes: C6140D (High level languages); C6110V (Visual programming); C6115 (Programming support); C6130B (Graphics techniques); C6110F (Formal methods)

Copyright 1995, IEE

16/5/19 (Item 7 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

05601794 INSPEC Abstract Number: C9403-6180-023

Title: EmbeddedButtons: documents as user interfaces

Author(s): Bier, E.A.

Author Affiliation: Xerox Palo Alto Res. Center, CA, USA

Conference Title: UIST Fourth Annual Symposium on User Interface Software and Technology. Proceedings of the ACM Symposium on User Interface Software and Technology p.45-53

Publisher: ACM, New York, NY, USA

Publication Date: 1991 Country of Publication: USA 232 pp.

ISBN: 0 89791 451 1

U.S. Copyright Clearance Center Code: 0 89791 451 1/91/0010/0045\$1.50

Conference Sponsor: ACM; SIGGRAPH; SIGCHI

Conference Date: 11-13 Nov. 1991 Conference Location: Hilton Head, SC, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: Recent electronic document editors and hypertext systems allow users to create customized user interfaces by adding user-pressable buttons to on-screen documents. Positioning these buttons is easy because users are already familiar with the use of document editors. Unfortunately, the resulting user interfaces often exist only in stand-alone document systems, making it hard to integrate them with other **applications**. Furthermore, because buttons are usually treated as special document **objects**, they cannot take advantage of document editor formatting and layout capabilities to create their appearance. This paper describes the EmbeddedButtons architecture, which makes it easy to integrate buttons into documents and to use the resulting documents for a variety of user interface types. EmbeddedButtons allows arbitrary document elements to behave as buttons. Documents can be **linked** to **application windows** to serve as **application** control panels. Buttons can store and display **application** state to serve as mode indicators. **New** button classes, editors, and **applications** can be added dynamically. (23 Refs)

Subfile: C

Descriptors: document handling; human factors; hypermedia; text editing; user interfaces

Identifiers: EmbeddedButtons; electronic document editors; user interfaces; hypertext systems; customized user interfaces; user-pressable buttons; on-screen documents; document editor formatting; document layout capabilities; **application** windows; **application** control panels; button classes

Class Codes: C6180 (User interfaces); C6160Z (Other DBMS); C6130D (Document processing techniques)

16/5/20 (Item 8 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

04819750 INSPEC Abstract Number: A91026152, C91018178

Title: VideoGraph: a new way to study kinematics

Author(s): Beichner, R.J.; DeMarco, M.J.; Ettestad, D.J.; Gleason, E.

Author Affiliation: Center for Learning & Technol., State Univ. of New York, Buffalo, NY, USA

Conference Title: Conference on Computers in Physics Instruction. Proceedings p.244-5

Editor(s): Redish, E.F.; Risley, J.S.

Publisher: Addison-Wesley, Redwood City, CA, USA

Publication Date: 1990 Country of Publication: USA xix+586 pp.

ISBN: 0 201 16306 3

Conference Sponsor: AIP; APS; NSF; IBM; et al

Conference Date: 1-5 Aug. 1988 Conference Location: Raleigh, NC, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Applications (A); Practical (P); Experimental (X)

Abstract: Summary form only given. The VideoGraph **software** package provides students with a **new** way of taking data on the motion of **objects**. Targeted for any introductory high school or college-level physics course, this **software** not only makes it easier and faster to collect motion data, but also helps students make the cognitive link between the physical event and the mathematical graph representing it. Students begin by videotaping an interesting motion. The VideoGraph **software** presents

each frame individually. The student moves a mouse-controlled cursor until it is on a readily discernible part of the moving **object** . Clicking the mouse button records the position of that point and automatically advances to the **next frame** . This process continues until all frames have been marked. It generally takes fewer than two seconds per frame. Now the student selects a graph from the 'Windows' menu. For example, if the student selects the 'X-Position' graph, the **software** opens a **new window** and draws appropriate axes. Selecting 'Animate' under the 'View' menu then produces a movie showing the **object** as it goes across the screen. At the same time the **software** generates a graph of the **object** 's position along the x direction. It can also produce graphs of position, speed and acceleration for both x and y directions or can display a list of coordinates for the graph. (3 Refs)

Subfile: A C

Descriptors: computer graphics; educational computing; kinematics; physics computing; **software** packages; user interfaces

Identifiers: high school physics; kinematics; VideoGraph **software** package; college-level physics course; physical event; mathematical graph; videotaping; mouse-controlled cursor; moving **object** ; position; speed; acceleration; coordinates

Class Codes: A0150H (Instructional computer use); A0320 (Classical mechanics of discrete systems: general mathematical aspects); A0630 (Measurement of basic variables); A0150P (Laboratory experiments and apparatus); A0650 (Data handling and computation); C7320 (Physics and Chemistry); C5260B (Computer vision and picture processing); C6130B (Graphics techniques); C7810C (Computer-aided instruction)

16/5/21 (Item 9 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

04181745 INSPEC Abstract Number: C88046566

Title: WATCOM C 6.0 (compiler)

Author(s): Hisley, P.N.

Journal: PC Tech Journal vol.6, no.6 p.140-5

Publication Date: June 1988 Country of Publication: USA

CODEN: PCTJEZ ISSN: 0738-0194

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P); Product Review (R)

Abstract: WATCOM has produced the C 6.0 language package. It includes the WATCOM C 6.0 optimizing compiler, the Express C compiler with integrated development environment, a windowed source-code level debugger, a **screen editor**, an overlay **linker**, a library manager, Make and Touch utilities, and an **object** disassembler. A full-bore optimizing C compiler, WATCOM's C 6.0 performs excellent code optimization. The compiler effectively uses the limited 80*86 register set and addresses the problem of **function call** overhead by passing as many **function** parameters as possible in registers. This approach minimizes the overhead incurred by pushing parameters onto the stack prior to a **function call** and restoring the stack upon **function** return. (0 Refs)

Subfile: C

Descriptors: C language; microcomputer **applications** ; **program** compilers; **software** packages

Identifiers: WATCOM products; Make utility; ANSI-compatible; language package; WATCOM C; optimizing compiler; Express C compiler; integrated development environment; windowed source-code level debugger; screen editor ; overlay linker; library manager; Touch utilities; **object** disassembler; full-bore optimizing C compiler; code optimization; limited 80*86 register set; **function call** overhead; function parameters; registers; stack

Class Codes: C6140D (High level languages); C6150C (Compilers, interpreters and other processors)

16/5/25 (Item 1 from file: 99)

DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs
(c) 2006 The HW Wilson Co. All rts. reserv.

1713228 H.W. WILSON RECORD NUMBER: BAST93021652

Microsoft demonstrates OLE 2.0

Andrews, Dave;

Byte v. 18 (Spring '93) p. 22

DOCUMENT TYPE: Feature Article ISSN: 0360-5280 LANGUAGE: English

RECORD STATUS: Corrected or revised record

ABSTRACT: Part of a special issue on Windows. Microsoft is about to release the final **software** development kit for OLE 2.0 for 16-bit Windows, which will allow developers to add **object linking** and embedding (OLE) capability to **Windows applications**. With OLE, a word processing **program** can contain a live **object**, such as a spreadsheet chart, that was created in another **program**. Changes made in the spreadsheet **program** automatically become **updated** in the word processing **program** as well. .

DESCRIPTORS: OLE (Computer **programs**); Windows (Computer **programs**);
Microsoft Corporation;

16/5/29 (Item 3 from file: 95)

DIALOG(R)File 95:TEME-Technology & Management
(c) 2006 FIZ TECHNIK. All rts. reserv.

00730813 E93124481010

Software development on the basis of frame-channel model

(Softwareentwicklung auf der Basis des Frame-Channel-Modells)

Maurer, H; Scherbakov, N

TU Graz, A

SPSE 92, Shifting Paradigms in Software Engineering, Proc. of the 7th Joint Conf. of OCG and NJSZT, Klagenfurt, A, Sep. 21-23, 19921992

Document type: Conference paper Language: English

Record type: Abstract

ISBN: 3-211-82408-1; 0-387-82408-1

ABSTRACT:

In the article, a **new** paradigm in **software** engineering is discussed. In accordance with this paradigm a **software** system can be seen as a number of so-called **frames connected** by a number of channels. Hence, the authors call this model the frame-channel model. Frames can encapsulate concrete actions such as execution of procedures, interpretation of database queries, infer procedures, and so on. The concept of channels allows to combine a number of frames into single **software** system in an elegant fashion. The model can also be used in coauthoring numerous, large, **software** -related documents throughout the **software** life cycle.

DESCRIPTORS: **PROGRAM** DEVELOPMENT; THEORETICAL MODELS; **PROGRAM** STRUCTURE
; **OBJECT** ORIENTED PROGRAMMING; INPUT DATA; OUTPUT DATA

IDENTIFIERS: Softwareentwicklung; frame-channel

16/5/30 (Item 4 from file: 95)

DIALOG(R)File 95:TEME-Technology & Management
(c) 2006 FIZ TECHNIK. All rts. reserv.

00689556 E93073584020.

Creating objects for the workplace shell

(Zur Erzeugung von Objekten fuer die Workplace Shell)

Coulombe, J

IBM; Cary, USA

SHARE Europe Anniversary Meeting, Proc. Davos, CH, Sep. 28 - Oct. 2, 1992
1992

Document type: Conference paper Language: English

Record type: Abstract

ABSTRACT:

An important part of the announcement of OS/2 2.0 in the Spring of 1992 was its new user environment, called the Workplace. This user environment is completely **object** oriented ,yet allows for backward compatibility with existing DOS, OS/2 1.X and **Windows 3.X applications** . In order to develop **new programs** that take advantage of the Workplace shell and that are fully integrated with it, developers must learn a few new tricks and add a couple of steps to their development process. This paper describes the design strategy and the steps necessary to develop **objects** for the Workplace Shell.

DESCRIPTORS: INTERNATIONAL BUSINESS MACHINES CORPORATION; **OBJECT** ORIENTED PROGRAMMING; USER INTERFACES; WINDOW SYSTEM; CONVERSATIONAL SYSTEMS; MAN MACHINE SYSTEMS; GRAPHIC PRESENTATION; SYSTEM DESCRIPTION; USER FRIENDLINESS; **SOFTWARE** TOOLS; **PROGRAM** DEVELOPMENT; PROGRAMMING SYSTEM; PORTABILITY-- **SOFTWARE** ; IMPLEMENTATION; JOB SHOP; MICROCOMPUTERS; CONTROL PANELS; **PROGRAM** FETCH; OS 2--OPERATING SYSTEM

IDENTIFIERS: X 11 WINDOW SYSTEM; Objekterzeugung; Arbeitsplatz-Shell

?

25/5/2 (Item 2 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04841551 E.I. No: EIP97103851835

Title: Aggregates: Using design patterns to create implicitly parallel data structures in C plus plus

Author: Hudak, David E.; Baughman, Nathan; Hodges, Greg

Corporate Source: Ohio Northern Univ, Ada, OH, USA

Conference Title: Proceedings of the 1997 IEEE National Aerospace and Electronics Conference, NAECON. Part 1 (of 2)

Conference Location: Dayton, OH, USA Conference Date: 19970714-19970717

Sponsor: IEEE

E.I. Conference No.: 47074

Source: National Aerospace and Electronics Conference, Proceedings of the IEEE v 1 1997. IEEE, Piscataway, NJ, USA, 97CH36015. p 239-246

Publication Year: 1997

CODEN: NASEA9

Language: English

Document Type: CA; (Conference Article) Treatment: A; (Applications)

Journal Announcement: 9711W4

Abstract: Small-scale parallel platforms have become prevalent in the marketplace. These machines feature globally shared address spaces, complex kernels and an emphasis on throughput-based parallelism. These new platforms require **new** tools for concurrent programming. **Programs** written for these **new** platforms must be efficient, portable and adaptable to differing machine conditions. In addition, programming environments for these new platforms should integrate concurrency into the design process in order to exploit the multiple processors for speedup when sufficient workloads arise. This functionality can be effectively embedded at the framework/API level. Efficient design of these features can be facilitated using **object** -oriented design techniques known as design patterns. To illustrate these concepts, we developed a framework called aggregates. Aggregates are a suggested replacement for arrays, linked lists, or other structures for holding large collections of C plus plus **objects**. Aggregates use run-time partitioning to support concurrent **application** of a member function to all **objects** in the aggregate. Experimental results on a single-processor **Windows** NT system and a multiprocessor Silicon

Graphics Power Challenge have demonstrated a single **program** using aggregates performed comparably to traditional arrays and linked lists in a single processor environment while providing speedup in a multiple processor environment. (Author abstract) 9 Refs.

Descriptors: *Parallel processing systems; Computer architecture; C (programming language); **Object** oriented programming; Membership functions; Data structures

Identifiers: Run time partitioning

Classification Codes:

723.1.1 (Computer Programming Languages)

722.4 (Digital Computers & Systems); 723.1 (Computer Programming)

722 (Computer Hardware); 723 (Computer Software); 921 (Applied Mathematics)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

25/5/3 (Item 3 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04518854 E.I. No: EIP96103356409

Title: FrameView: Object -oriented visualization system for frame

analysis

Author: Moni, Sheloney; White, Donald W.

Corporate Source: Purdue Univ, West Lafayette, IN, USA

Source: Journal of Computing in Civil Engineering v 10 n 4 Oct 1996. p 276-285

Publication Year: 1996

CODEN: JCCEE5 ISSN: 0887-3801

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); G; (General Review)

Journal Announcement: 9612W1

Abstract: Rapidly developing desktop computing capabilities, which include high resolution graphics and interactive graphical user interfaces, are leading to a **new** generation of engineering **software**. One of the challenges in engineering **software** development is the effective use of computer graphics for visualization of data. **Object** -oriented methodologies hold the greatest promise to support reuse and rapid prototyping for large-scale **software** development. This paper outlines some of the essential characteristics of FrameView, an **object** -oriented **software** system for visualization of responses from any type of frame analysis model. FrameView provides a graphical user interface that is based on the X- **Window** system and Motif. The **graphics** library used is PEX, which is a three-dimensional (3D) extension to the X-Window system. The programming language used is C plus plus. Important attributes of PEX, and tedious programming details required when using PEX, have been encapsulated in a number of classes. General classes are developed to abstract the tools needed for viewing frame-element responses and two-dimensional (2D) and 3D **graphics** modeling. Viewing of results from 2D **frame** elements based on an assumed cubic transverse displacement field for drawing the deflected shape, and an assumed linear function for any other general response quantities along the element length, have been incorporated in FrameView. An example of how FrameView has been extended to handle other graphical representations of frame-element responses is discussed. (Author abstract) 21 Refs.

Descriptors: *Structural **frames**; **Object** oriented programming; Three dimensional computer **graphics**; Graphical user interfaces; C (programming language); Personal computers; Finite element method; Structural analysis; Computer simulation; Computer **software**

Identifiers: **Software** package FrameView; High resolution **graphics**; **Frame** analysis model; **Software** package X-Window; Software package Motif

Classification Codes:

723.1.1 (Computer Programming Languages)

408.2 (Structural Members & Shapes); 723.1 (Computer Programming);

723.5 (Computer Applications); 722.2 (Computer Peripheral Equipment);

722.4 (Digital Computers & Systems)

408 (Structural Design); 723 (Computer Software); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING)

25/5/4 (Item 4 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

04335742 E.I. No: EIP96013014158

Title: Project object

Author: Shane, Richard M.; Zagona, Edith A.; McIntosh, Dave; Fulp, Terrance J.; Goranflo, H. Morgan

Corporate Source: Tennessee Valley Authority, Knoxville, TN, USA

Source: Civil Engineering (New York) v 66 n 1 Jan 1996. p 61-63

Publication Year: 1996

CODEN: CIEGAG ISSN: 0885-7024

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); G; (General Review)

Journal Announcement: 9603W4

Abstract: Modeling and managing water systems is a notoriously difficult and complex task. However, they are made simpler by PRYSM, a **new object**-oriented **software** package which provides a convenient modeling structure to plan for optimal hydroenergy production from reservoir system with constantly changing nonpower objectives and environmental constraints.

Object libraries provided with PRYSM contain many modeling methods suitable for general use.

Descriptors: *Water resources; Computer **software** ; Management; C (programming language); Computer simulation; Reservoirs (water); Power plants; Rivers; River diversion; Canals

Identifiers: Water system management; **Object** oriented **software** package; River reaches; River confluences; Icon programming; **Screen** palette; **Graphic** workspace; Controller **object** ; Controller library

Classification Codes:

723.1.1 (Computer Programming Languages)

912.2 (Management); 723.1 (Computer Programming); 723.5 (Computer Applications); 441.2 (Reservoirs)

444 (Water Resources); 723 (Computer Software); 912 (Industrial Engineering & Management); 441 (Dams & Reservoirs)

44 (WATER & WATERWORKS ENGINEERING); 72 (COMPUTERS & DATA PROCESSING); 91 (ENGINEERING MANAGEMENT)

25/5/5 (Item 5 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

03106210 E.I. Monthly No: EIM9108-038240

Title: BASEstar for VMS.

Author: Kinsey, G.

Corporate Source: Digital Equipment GmbH, Ger

Conference Title: Proceedings of the 23rd International Symposium on Automotive Technology and Automation

Conference Location: Vienna, Austria Conference Date: 19901203

E.I. Conference No.: 14327

Source: Advanced Automotive Manufacturing Proceedings - International Symposium on Automotive Technology & Automation v 3. Publ by Automotive Automation Ltd, Croydon, Engl. p 274-278

Publication Year: 1990

CODEN: PISADA

Language: English

Document Type: PA; (Conference Paper) Treatment: A; (Applications)

Journal Announcement: 9108

Abstract: BASEstar for VMS is Digital's foundation for creating integrated solutions for an industry in which each customer has a unique and complex set of needs. BASEstar allows manufacturers to have a foundation for creating their integrated manufacturing solutions using a building-block approach. The BASEstar platform integrates data from **applications** and devices currently in use in the manufacturing environment while facilitating the creation of **new applications** for sharing factory data. **Application software** layered on the BASEstar platform is independent of actual data sources such as **applications** and factory devices. **Software** is adapted easily by changing definitions and not the

applications themselves. This approach allows **new** technologies to be incorporated into solutions. A graphical interface provides a development system for building and managing **graphics screens** using an **object**-oriented architecture.

Descriptors: *COMPUTER INTEGRATED MANUFACTURING--*Interfaces; COMPUTER GRAPHICS; COMPUTER INTERFACES--Reviews; COMPUTER **SOFTWARE** --Performance; DATA PROCESSING, BUSINESS

Identifiers: MANUFACTURING MESSAGE SPECIFICATION (MMS); **SOFTWARE** PACKAGE BASESTAR

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

25/5/6 (Item 1 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2006 ProQuest Info&Learning. All rts. reserv.

01233298 ORDER NO: AAD92-20907

PAVEMENT MANAGEMENT DECISION METHODOLOGY AND DECISION SUPPORT SYSTEM

Author: SHEN, YUNG-CHING

Degree: PH.D.

Year: 1991

Corporate Source/Institution: RENSSELAER POLYTECHNIC INSTITUTE (0185)

Adviser: DIMITRI A. GRIVAS

Source: VOLUME 53/03-B OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 1519. 228 PAGES

Descriptors: ENGINEERING, CIVIL; COMPUTER SCIENCE; BUSINESS
ADMINISTRATION, MANAGEMENT

Descriptor Codes: 0543; 0984; 0454

The primary theme of this study is the formalization of decision methodologies in a decision support system for pavement management. Specific methodologies incorporated in the system are: fuzzy set approach for damage assessment and knowledge graph concept for decision-making processes.

The magnitude of structural damage is established using linguistic scales that have been formulated on the basis of pattern classification. Based on fuzzy set theory, linguistic distress values are represented by membership functions and a single measure of overall pavement damage is determined through fuzzy weighted average computation and linguistic approximation. The concept of knowledge graph is applied to formalize two ill-structured tasks: condition diagnosis and treatment identification. The evolution of knowledge graphs consists of problem decomposition, term interpretation, and heuristic organization.

The representation schemes used to organize the knowledge and data constitute the architecture of the decision support system, which consists of a knowledge-based system and a database system. Knowledge **graphs**, coupled with a **frame** representation scheme, have become the basis for developing a prototype knowledge base. The prototype system has been implemented within an expert system shell, Nexpert **Object**.

The functional specifications of a pavement integrated database system (PIDBS) are formalized. The system analysis follows a top-down approach that encompasses all functions within the pavement management process. The design of the database utilizes the "Entity-Relationship" data modeling technique. PIDBS has been implemented using the ORACLE relational database management system. **Applications** include interactive interfaces for menus, forms, and reports, and programmatic interfaces for damage analysis and the knowledge-based system.

This study presents illustrative examples to demonstrate the use of

knowledge base, database, and damage assessment in the **New** York State Thruway pavement maintenance **applications** . System integration and the developed methodologies in providing assistance for data and decision needs are discussed. This study concludes that the formalization of decision methodologies and the functional integrity of the decision support system are the most important contributions to the system development.

25/5/11 (Item 5 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

06287773 INSPEC Abstract Number: C9607-6180G-032

Title: A construction method and application of object-oriented widgets using a non-object-oriented language

Author(s): Ok-Gee Min; Chi-Jung Hwang

Journal: Journal of KISS(C) (Computing Practices) vol.2, no.1 p. 38-45

Publisher: Korea Inf. Sci. Soc,

Publication Date: March 1996 Country of Publication: South Korea

CODEN: CKNCFY

Material Identity Number: E347-96003

Language: Korean Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: In this paper, an **object** -oriented widget, which satisfies the main characteristics of **object** -orientedness such as encapsulation, inheritance, and polymorphism, is constructed by using a non-**object** oriented language C. **Object** -oriented widgets are effectively applicable to the construction of **graphic** user interfaces using **X Window** . A new framework model, an **application** of the **object** -oriented widgets, can be constructed by adding the **object** -oriented widgets to a conventional framework model. A framework model proposed is the one through which a programmer can use both frames and the toolbox. Such a framework model allows us to have system-wide consistency and high extensibility by the notion of inheritance, and it is independent on languages in the sense that it employs a widely used language. (11 Refs)

Subfile: C

Descriptors: C language; data encapsulation; graphical user interfaces; inheritance; **object** -oriented methods

Identifiers: **object** -oriented widgets; encapsulation; inheritance; polymorphism; C; **graphic** user interfaces; **X Window**; framework model; frames; toolbox

Class Codes: C6180G (Graphical user interfaces); C6140D (High level languages); C6110J (Object-oriented programming)

Copyright 1996, IEE

25/5/12 (Item 6 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

06190146 INSPEC Abstract Number: C9603-6130D-031

Title: On computerisation and representation of characters

Author(s): Hussain, F.

Author Affiliation: Dept. of Comput. & Inf. Sci., De Montfort Univ., Milton Keynes, UK

Conference Title: IEE Colloquium on Document Image Processing and Multimedia Environments (Digest No.1995/191) p.11/1-9

Publisher: IEE, London, UK

Publication Date: 1995 Country of Publication: UK 118 pp.

Material Identity Number: XX96-00284

Conference Title: IEE Colloquium on Document Image Processing and Multimedia Environments (Digest No.1995/191)

Conference Sponsor: IEE

Conference Date: 2 Nov. 1995 Conference Location: London, UK

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P); Theoretical (T)

Abstract: The commercial need to computerise contours of **objects** has intensified over **recent** years as more and more **applications** endeavour to seek the benefits of visualisation. The primary aim has been to gain a computerised model which closely resembles the original shape and form. The digital form lends itself to be processed by fast processors, and their output to be displayed on **graphic** devices such as **screen** displays and laser printers. The benefits of the digital form are also extended to cater for numerically controlled machines. These are employed to produce outlines through cutting or engraving on paper, metal, plastic or wood. When it comes to digitising the outlines of characters, additional considerations have to be included in the modelling, design, phase. The aim of these is to capture the uniqueness and the distinct features of a font, which traditionally were realised by a proficient artisan's hand and eye. Modern approaches endeavour to gain the same impact by using a set of pre-defined specifications in the design process. The goal of which is not just to model accurately a desired font, but also to ensure that the displayed version is both legible and aesthetically acceptable. This paper provides an insight about the design methods used to both model and represent the contours of Latin characters and fonts. It describes the role of the spline as a means of realising the modelling aspect. To this regard, the spline descriptions used by Postscript (Bezier cubic) and TrueType (parabolic) are looked at, and a comparative analysis made. The discussion throughout is levelled at providing a practical understanding of the subject matter. (17 Refs)

Subfile: C

Descriptors: document handling; document image processing; image representation; optical character recognition

Identifiers: OCR; optical character recognition; image representation; computerisation; character representation; **object** contour; document handling; computerised model; visualisation; Latin letter; spline; Postscript; Bezier cubic; TrueType; parabolic

Class Codes: C6130D (Document processing techniques); C1250B (Character recognition); C5260B (Computer vision and image processing techniques)

Copyright 1996, IEE

25/5/13 (Item 7 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

04859695 INSPEC Abstract Number: C91027157

Title: Fine tuning expands the bounds of data acquisition

Author(s): Labs, W.

Journal: I&CS vol.64, no.1 p.23-8

Publication Date: Jan. 1991 Country of Publication: USA

CODEN: CHISDY ISSN: 0746-2395

Language: English Document Type: Journal Paper (JP)

Treatment: General, Review (G); Practical (P)

Abstract: Data acquisition hardware and **software** keep getting better. Sampling speed and throughput to host computer memory have been increased by digital signal processing (DSP) techniques, fast microprocessors (mu Ps), and direct memory access. Improvements in **software** include icon-driven menus, virtual instrument screens, 32-bit **application**

software processing, SPC/SQC and other **software** techniques that help engineers get useful information out of collected data, improved **graphics** user interfaces (GUIs) such as **Windows** 3.0 and OS/2-Presentation Manager, **software** libraries, **software** driver toolkits, and a host of **new** **application** products. The author discusses the very latest trends in data acquisition systems (DASs). (0 Refs)

Subfile: C

Descriptors: data acquisition; digital signal processing chips; graphical user interfaces

Identifiers: **object** -oriented programming; data acquisition; host computer memory; digital signal processing; fast microprocessors; direct memory access; icon-driven menus; virtual instrument screens; **application software** ; SPC/SQC; graphics user interfaces; Windows 3.0; OS/2-Presentation Manager; **software** libraries; **software** driver toolkits ; 32 bit

Class Codes: C3210G (Data acquisition systems); C5520 (Data acquisition equipment and techniques)

Numerical Indexing: word length 3.2E+01 bit

25/5/20 (Item 2 from file: 34)

DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

(c) 2006 Inst for Sci Info. All rts. reserv.

02793363 Genuine Article#: ME484 Number of References: 19

Title: AUTOMATED FLOW GRAPH-BASED TESTING OF OBJECT -ORIENTED SOFTWARE MODULES

Author(s): PARRISH AS; BORIE RB; CORDES DW

Corporate Source: UNIV ALABAMA, DEPT COMP SCI/TUSCALOOSA//AL/35487

Journal: JOURNAL OF SYSTEMS AND SOFTWARE, 1993, V23, N2 (NOV), P95-109

ISSN: 0164-1212

Language: ENGLISH Document Type: ARTICLE

Geographic Location: USA

Subfile: SciSearch

Journal Subject Category: COMPUTER APPLICATIONS & CYBERNETICS

Abstract: Classes represent the fundamental building blocks in **object** -oriented **software** development. Several techniques have been proposed for testing classes. However, most of these techniques are heavily specification based, in the sense that they demand the existence of formal specifications for the module. In addition, most existing techniques generate test cases at random rather than systematically. We present some test case generation techniques that are based entirely on class implementation, involve systematic generation of test cases, and are fully automated. Our techniques are based on an adaptation of existing white- **box** , flow **graph** -based techniques for unit testing conventional procedures and functions. We also provide a general conceptual framework to support the modeling of classes using flow graphs. Our framework clarifies the fundamental definitions and concepts associated with this method for modeling classes.

Identifiers--KeyWords Plus: SPECIFICATIONS

Research Fronts: 91-0090 001 (**OBJECT** -ORIENTED DATA MODELS; MULTIDATABASE SYSTEMS; INNOVATIVE **SOFTWARE** ENGINEERING ENVIRONMENTS)

91-2960 001 (REUSABLE **SOFTWARE** COMPONENTS; SYSTEMS IN ADA; **OBJECT** -ORIENTED DESIGN)

91-2961 001 (**OBJECT** -ORIENTED PROGRAMMING; LOGIC-BASED REQUIREMENT LANGUAGE FOR **NEW** **SOFTWARE** ENGINEERING PARADIGMS; ADA CODES)

Cited References:

BERARD EV, 1987, CREATING REUSABLE AD

BOOCH G, 1987, SOFTWARE COMPONENTS

BOUGE L, 1986, V6, P343, J SYST SOFTWARE

T/5/5-8

31/5/5 (Item 1 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

06715776 INSPEC Abstract Number: C9711-6110J-053

Title: Object oriented development on UNIX and NT platforms**Author(s):** Lumish, P.**Conference Title:** OOP'97. Conference Proceedings p.329-35**Publisher:** SIGS Conferences, Bergisch Gladbach, Germany**Publication Date:** 1997 **Country of Publication:** Germany 396 pp.**Material Identity Number:** XX97-00547**Conference Title:** Proceedings of OOP'97. Objekt orientiertes Programmieren**Conference Date:** 3-7 Feb. 1997 **Conference Location:** Muchen, Germany**Language:** English **Document Type:** Conference Paper (PA)**Treatment:** General, Review (G)

Abstract: All indications that can be inferred from all of the various activities underway in all facets of the computing industry indicate that it will be at least five years or more before the Intel platforms will outperform RISC based architectures. This means that there is no reason to consider bringing in Intel systems to replace your UNIX development machines for at least that period of time. You will, however, need to deliver **new applications** or rewritten **older applications** that were UNIX specific into **Windows** /Intel systems. One phenomenon to watch is how UNIX-exact NT becomes over the next few years since this will have impact on the complexity of delivering code for **applications** that will run on these systems. (0 Refs)

Subfile: C**Descriptors:** network operating systems; **object** -oriented methods;**object** -oriented programming; Unix**Identifiers:** **object** oriented development; UNIX platforms; NT platforms; Intel platforms; UNIX development machines**Class Codes:** C6110J (Object-oriented programming); C6110F (Formal methods); C6150J (Operating systems); C6150N (Distributed systems software)

Copyright 1997, IEE

31/5/6 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

06432872 INSPEC Abstract Number: C9701-6110B-034

Title: Putting the RAPID into RAD**Author(s):** Smit, T.**Author Affiliation:** IBM Corp., Research Triangle Park, NC, USA**Journal:** American Programmer vol.9, no.11 p.19-24**Publisher:** Cutter Inf. Corp,**Publication Date:** Nov. 1996 **Country of Publication:** USA**ISSN:** 1048-5600**SICI:** 1048-5600(199611)9:11L:19:PRI;1-9**Material Identity Number:** D397-96011**Language:** English **Document Type:** Journal Paper (JP)**Treatment:** Practical (P)

Abstract: Even in today's emerging client-server/ **object** -oriented world, previous successful coding techniques are cloned and modified to accomplish the **new** task. The **old program** becomes a template for the **new program**, jump-starting the **application** development process. This type of code reuse is the basis for templating technology. Typically if a rapid

application development (RAD) approach is used to develop a business system, work begins with gathering end-user requirements. The end users need to see a vision of the system that is to be created. This vision represents the front face, or the facade, of the RAD Cube. The use of a GUI- or **screen** -building tool to prototype the interface, in conjunction with a joint **application** design (JAD) session, provides a quick identification of the business requirements and business justification. The author considers how templates can help the RAD process. She considers BW/Wizard, from Bridgewater Consultants Inc., that automates the process of taking the industrial strength model code and modifying it to meet your **new application** database access requirements. (0 Refs)

Subfile: C

Descriptors: business data processing; database management systems;

software reusability; **software** tools; visual programming

Identifiers: client-server systems; **object** -oriented systems; **program** template; **software** reuse; rapid **application** development; business system; end-user requirements; GUI; **screen** -building tool; joint **application** design; business requirements; BW/Wizard; Bridgewater Consultants; **application** database access requirements; graphical user interface; visual programming

Class Codes: C6110B (Software engineering techniques); C6110V (Visual programming); C6115 (Programming support)

Copyright 1996, IEE

31/5/7 (Item 3 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

05338155 INSPEC Abstract Number: C9303-6115-061

Title: Completing the job interface design

Author(s): Rudolf, J.; Waite, C.

Journal: IEEE Software vol.9, no.6 p.11-22

Publication Date: Nov. 1992 Country of Publication: USA

CODEN: IESOEI ISSN: 0740-7459

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: HyperNews, a user-interface management system that lets a user separate **application** and interface design to link a **new** interface to an **old application** with very little programming, is described. Much of a HyperNews interface can be designed without writing any code at all. With direct manipulation, a user can design a graphical user interface simply by creating, moving, and resizing **objects** on the **screen**. Experimentation with different interface styles is possible with less effort in HyperNews compared to specification languages and tool kits, and immediate feedback can result in extremely fast development. The ways HyperNews provides communication capabilities among all HyperNews **objects** (such as control **objects**, cards, backgrounds, and stacks) and links interfaces to internal **applications**, which were developed within HyperNews, to external **applications**, or to existing **applications** developed without HyperNews are discussed. (8 Refs)

Subfile: C

Descriptors: graphical user interfaces; hypermedia; user interface management systems

Identifiers: **object** creation; **object** moving; **object** resizing; job interface design; HyperNews; user-interface management system; direct manipulation; graphical user interface; interface styles; control **objects**; cards; backgrounds; stacks

Class Codes: C6115 (Programming support); C6180G (Graphical user interfaces); C6130B (Graphics techniques); C6160Z (Other DBMS)

31/5/8 (Item 1 from file: 266)

DIALOG(R) File 266:FEDRIP

Comp & dist by NTIS, Intl Copyright All Rights Res. All rts. reserv.

00507955

IDENTIFYING NO.: 31-0642 AGENCY CODE: SBIR

INNOVATIVE OPERATING SOFTWARE

PRINCIPAL INVESTIGATOR: hemaly, A. resident

PERFORMING ORG.: ECHNOSOFT, INC., 1180 Reed Hartman Highway, INCINNATI, H 5242

FY : 2000 FUNDS: \$9,954 (9000) TYPE OF AWARD: 0016403C6018

SUMMARY: roposed herein is a collaborative engineering environment based on an object-oriented, web-enabled, multidisciplinary, distributed computing framework supporting **application** development for integrated product and process engineering. It supports specification of **software** elements through UML diagrams and detailed graphical composition of **object** hierarchies and process schematics for comprehensive **application** development. It provides a visual environment supporting **object**-oriented application development, user interface customization, legacy tool integration, and knowledge base population. **Legacy software** tools can be **updated** into **object** structures and associated with newly defined **objects** for composing and developing large scale **applications**. It facilitates distributed **object** computing enabling the interoperability among distributed **applications** on a network of heterogeneous computers. It supports concurrent engineering through a web-enabled environment linking multiple users collaborating in real time in the engineering process. Classes, models, and process structure can be organized within knowledge bases providing support for application development. Model patterns can be organized in knowledge bases providing the foundation for knowledge management and information processing for design exploration and trade studies. It provides a comprehensive visual environment for **software** engineering and development, and incorporate utilities for packaging and deployment of multidisciplinary application. Ultimately, it reduces the time and cost of custom **application** development and maintenance from concept to deployment. The development of the proposed architecture will result in an open, visual and modular collaborative engineering framework with an underlying **object**-oriented architecture. The framework employs a unique and innovative approach to product and process **application** engineering, development, deployment, and maintenance. It will provide maximum interoperability among application **program**. The proposed framework's web-enabled modeling environment will facilitate real-time collaboration among participants, from various disciplines both within and outside of an organization and including various suppliers and consultants, all of whom will have the opportunity to engage and draw on each other's expertise. The successful development of this **framework** will reduce **application software** development time and cost, enabling quick response to the fast-paced development of new engineering system. The defense industry will benefit from the framework in the development of custom **applications** from concept to the validation stages, capitalizing on investment in legacy tools while adopting new methods for **updating** and development of **new applications**. The framework will also have a great impact on the various applications in the design-to-production automation of complex systems in the automotive and aerospace industries, as well as major equipment manufacturing. After a successful Phase II project TechnoSoft will productize the proposed framework and market it as a part of its product offering. The payoff for the US industry will be substantial resulting from rapid-to-market **software** deployment and reduced life-cycle cost. Together with its current customers TechnoSoft will deploy the

framework in the engineering of the DoD **applications** .
?

T/5/2,3,5,6,7-9,11,13,14,16,17,19,20

38/5/2 (Item 2 from file: 8)

DIALOG(R)File 8: Ei Compendex(R)

(c) 2006 Elsevier Eng. Info. Inc. All rts. reserv.

03950095 E.I. No: EIP94091407246

Title: Configurable systems for reactive production management

Author: Smith, Stephen F.; Lassila, Ora

Corporate Source: Carnegie Mellon Univ, Pittsburgh, PA, USA

Conference Title: Proceedings of the IFIP TC5/WG5.7 International Workshop on Knowledge-Based Reactive Scheduling

Conference Location: Athens, Greece Conference Date: 19931001

E.I. Conference No.: 20946

Source: IFIP Transactions B: Computer Applications in Technology n B-15 1994. p 93-106

Publication Year: 1994

CODEN: ITBTEH ISSN: 0926-5481 ISBN: 0-444-81814-6

Language: English

Document Type: MC; (Monograph Chapter) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9411W2

Abstract: In this paper, we outline current work into the development of reconfigurable scheduling systems: systems that enable rapid customization to specific production environments through encapsulation, extension, and reuse of component scheduling 'services' (e.g., domain modeling primitives, constraint management and analysis techniques, solution subprocedures, problem decomposition and configuration heuristics). Our approach is grounded on two basic premises: (1) that system organization and decision-support 'services' should directly reflect the inherently reactive nature of decision-making in practical scheduling environments, and (2) that diversity in the character and requirements of different **application** environments will invariably require different specialized scheduling support services. We adopt a general constraint-based model of scheduling as an iterative, opportunistic process of schedule revision, which provides an architectural framework for formulating the configuration problem, and utilize **object** programming techniques to compositionally construct component services. Our overall goal is the development of an **application** building environment, which combines a 'tool box' of basic modeling and scheduling primitives with facilities for assembling, aggregating and specializing these primitives to define the decision support functionality (or services) required in a given **application** context. As **new** services are composed, they can be encapsulated as additional, higher level tools and are available for **reuse** in subsequent **applications**. (Author abstract) 26 Refs.

Descriptors: *Production control; **Object** oriented programming; Artificial intelligence; Expert systems; Heuristic methods; Mathematical models; Constraint theory; Decision support systems; Scheduling

Identifiers: Reconfigurable scheduling systems; Reactive production management

Classification Codes:

723.4.1 (Expert Systems)

913.2 (Production Control); 723.1 (Computer Programming); 723.4 (Artificial Intelligence); 921.6 (Numerical Methods); 921.5 (Optimization Techniques)

913 (Production Planning & Control); 723 (Computer Software); 921 (Applied Mathematics)

91 (ENGINEERING MANAGEMENT); 72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

38/5/3 (Item 1 from file: 35)

DIALOG(R) File 35:Dissertation Abs Online
(c) 2006 ProQuest Info&Learning. All rts. reserv.

01493691 ORDER NO: AADAA-IMM05141

COMPLETENESS OF LARCH/C++ SPECIFICATIONS FOR BLACK- BOX REUSE

Author: UMANSKY, ILYA

Degree: M.C.S.

Year: 1995

Corporate Source/Institution: CONCORDIA UNIVERSITY (CANADA) (0228)

Adviser: V. S. ALAGAR

Source: VOLUME 34/04 of MASTERS ABSTRACTS.

PAGE 1622. 145 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

ISBN: 0-612-05141-2

The **object** -oriented paradigm introduced **new** capabilities for an effective **software** development. One of the most promising benefits is the possibility to **reuse** **software** components which were built and tested thoroughly. **Software reuse** is most effective when it is conducted in a black- **box** fashion. That is, when the **software** can be used without studying its source code or running tests to clarify its behavior. To achieve a black- **box reuse** the **reusable software** has to be complemented with its specification. Informal description is not adequate for this purpose because it lacks precision and does not have sufficient expressive power. As a result, researchers turn their attention toward formal specifications.

In this thesis we study completeness of the specifications for C++ classes intended for a black- **box** reuse. We present a definition of an interface behavior of a C++ class and completeness criteria for specifications to be able to convey this behavior. To provide practical means for the completeness verification we identify the completeness verification methodology and present the algorithm to apply this methodology.

The completeness verification algorithm developed in this thesis uses Larch Prover, an automatic proof assistant. We provide guidelines for using Larch Theorem prover when applying the completeness verification algorithm, and provide the means for incompleteness detection and localization as well as incompleteness correction. Finally, we generalize the completeness verification algorithm for C++ constructs having inheritance and virtual functions, and identify the cases when incompleteness can not be removed.

38/5/5 (Item 3 from file: 35)

DIALOG(R) File 35:Dissertation Abs Online
(c) 2006 ProQuest Info&Learning. All rts. reserv.

01395354 ORDER NO: AAD95-04137

SOFTWARE REUSE EXPERT ENVIRONMENT (SREE) (VIRTUAL REALITY)

Author: WANG, PETER HOR-CHING

Degree: PH.D.

Year: 1994

Corporate Source/Institution: UNIVERSITY OF ALABAMA IN HUNTSVILLE (0278)

Chairman: SAJJAN G. SHIVA

Source: VOLUME 55/09-B OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 3989. 150 PAGES

Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

The **Software Reuse** Expert Environment (SREE) assists the user in familiarizing himself with the domain **application** environment, locating partially-matched components from the reusable component library, understanding the life-cycle of components, and decomposing a component.

The knowledge-base was created by using the **frame** -based component representation method, and a set of decomposition rules and sibling rules. The component composition method is used to generate the **new software** system with minimum effort. The SREE is an open system which allows the user to modify the knowledge-base, the decomposition rules, and the sibling rules.

This research has focused on the knowledge-based approach with the combination of component composition and the **object** -oriented approaches. A prototype system has been designed for **software** development in the virtual reality **application** domain. In this system, a complete domain analysis is performed, and matching and component understanding mechanisms are provided.

38/5/6 (Item 4 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2006 ProQuest Info&Learning. All rts. reserv.

01294157 ORDER NO: AAD93-17692

RSS: AN APPROACH TO WIDESPREAD SOFTWARE REUSABILITY (REUSABILITY SUPPORT SYSTEM)

Author: KIM, YONGBEOM

Degree: PH.D.

Year: 1993

Corporate Source/Institution: NEW YORK UNIVERSITY, GRADUATE SCHOOL OF BUSINESS ADMINISTRATION (0868)

Source: VOLUME 54/02-A OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 590. 214 PAGES

Descriptors: BUSINESS ADMINISTRATION, MANAGEMENT; COMPUTER SCIENCE

Descriptor Codes: 0454; 0984

Software reuse refers to the use of previously developed **software** resources in **new applications**. In this dissertation, we concentrate on the concept of widespread reusability which means (1) **reuse** by other **software** developers within the same organization as well as the original developer, (2) reuse of data and process **objects** produced during the systems analysis and design phases as well as code, (3) **reuse** of general and specific purpose **software** resources across a variety of **application** domains, and (4) **reuse** of **software** resources along a continuum of task types from maintaining existing systems to developing **new software** systems.

The main objective of this dissertation is to search for and experiment with tools and techniques to support widespread **software reusability**. A survey of the literature was carried-out and it was concluded that achieving widespread **software reusability** would require a total systems approach and the development of specialized tools and techniques. This dissertation suggests extending the concept of CASE **software** towards a " **Reusability Support System**" (RSS) that provides automated support for the classification, storage, retrieval, and modification of **software** resources of all kinds.

A major impediment to **software reusability** is the difficulty of classifying various **software** resources in such a way that another person can easily identify them when they are needed to satisfy some new requirements. An important requirement for widespread reusability therefore is an effective means for classifying and retrieving **software** resources

from all phases of **software** system development.

To address this problem, three issues are investigated in this research: how to represent various **reusable software** resources for both classification and retrieval, how to organize a library of **reusable software** resources, and how to support the specification of user requirements.

A **frame** -based knowledge representation scheme employing concepts of abstraction is proposed to represent **reusable software** resources from all phases of the life cycle. A classification scheme based on the faceted method from library science is proposed. A **software** resource network, which is integrated into the faceted method, is used for organizing **software** resources. To provide more effective descriptive information, a "**software** resource model" (SRM) is proposed which emphasizes the explicit capture of semantic information as a means of increasing the precision of the classification- retrieval process. To demonstrate the feasibility and functionality of the proposed SRM approach, a prototype system was developed. In addition, an experiment testing the SRM approach against a keyword approach was carried-out using student subjects. The experiment suggests that the SRM approach can help in achieving more consistent classifications; however, the students found the approach quite difficult to understand and use. Improvements to the SRM model that may overcome these problems are suggested in the conclusion to the dissertation.

38/5/7 (Item 5 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2006 ProQuest Info&Learning. All rts. reserv.

01253508 ORDER NO: AAD92-35903

OBJECT -ORIENTED MODELING FOR INTEGRATED COMPUTER AIDED PROCESS
ENGINEERING: A SOFTWARE REUSE APPROACH

Author: MEHTA, JAIMIN A.

Degree: D.SC.

Year: 1992

Corporate Source/Institution: WASHINGTON UNIVERSITY (0252)

Adviser: R. L. MOTARD

Source: VOLUME 53/07-B OF DISSERTATION ABSTRACTS INTERNATIONAL.

PAGE 3645. 205 PAGES

Descriptors: ENGINEERING, CHEMICAL; COMPUTER SCIENCE

Descriptor Codes: 0542; 0984

There are two major problems in realizing Integrated Computer Aided Process Engineering (ICAPE) systems and environments: **object** -oriented modeling of process engineering data, and integration of the existing stock of **software** for process engineering. This research investigates a novel approach based on **software reuse** to solve both problems.

The main contribution of this research is a new , **software reuse** approach to **object** -oriented modeling for integration, and a systematic **software reuse** methodology called " **Reuse** for□**object**□ -orientation" or REO. The currently known **object** -oriented modeling methodologies prescribe development of a "universal" model for the **application** domain; thus they are practical only for new systems of limited scope. The REO methodology, on the contrary, provides a short-cut for deriving **object** -oriented models from the existing stock of **software** . The past and **current** research in **software** integration have examined the black **box** approach, wherein the tool is interfaced with its input and output only, and the glass **box** approach, wherein the tool is interfaced with its internal symbols, but used in as-is condition in its entirety. The REO methodology, on the contrary, provides an approach wherein only parts of a tool are used in an **object** -oriented system. Presently, the REO methodology covers two

software components: programming language descriptions and **program** descriptions.

The "experimental" subject includes parts of ASPEN, a chemical process modeling and simulation system, that is over a decade old and has over a quarter (1/4) million lines of **program** code. An **object** -oriented model is derived for this subject by following the REO methodology, and based on it a prototypical ICAPE system called "Icape-91" is designed and implemented in an experimental **object** -oriented system.

This research has identified and developed a novel approach to **software** integration and **object** -oriented modeling; an approach based on **software reuse** . **Software reuse** is a generalization of [software] integration. **Software reuse** can help in deriving **object** -oriented models from the existing stock of **software** . **Software reuse** can significantly assist **software** developers working in the field of ICAPE and ICAE in general. The successes of this research should motivate development, aided by REO, of large scale ICAPE systems or environments.

38/5/8 (Item 1 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

08067789 INSPEC Abstract Number: C2001-11-6130M-029

Title: Web design frameworks: an approach to improve reuse in Web applications

Author(s): Schwabe, D.; Rossi, G.; Esmeraldo, L.; Lyardet, F.

Author Affiliation: Dept. de Inf., PUC-Rio, Brazil

Book Title: Web engineerng. Managing diversity and complexity of web **application** development p.335-54

Editor(s): Murugesan, S.; Deshpande, Y.

Publisher: Springer-Verlag, Berlin, Germany

Country of Publication: Germany ix+355 pp.

ISBN: 3 540 42130 0 Material Identity Number: XB-2001-00042

Language: English Document Type: Book Chapter (BC)

Treatment: Practical (P)

Abstract: In this paper we introduce Web design frameworks as a conceptual approach to maximize **reuse** in Web **applications** . We first discuss the need for building abstract and reusable navigational design structures, exemplifying with different kinds of Web Information Systems. Then, we briefly review the state of the art of **object** -oriented **application** frameworks and present the rationale for a slightly different approach focusing on design reuse instead of code reuse. **Next** , we present OOHDM- **frame** , a syntax for defining the hot-spots of generic Web **application** designs. We illustrate the use of OOHDM-frame with a case study in the field of electronic commerce. We finally discuss how to implement Web design frameworks in different kind of Web platforms. (11 Refs)

Subfile: C

Descriptors: electronic commerce; electronic publishing; hypermedia; information resources; Internet; **object** -oriented programming

Identifiers: Web design frameworks; reuse; Web **applications** ; reusable navigational design structures; Web Information Systems; **object** -oriented **application** frameworks; design reuse; electronic commerce; Web platforms

Class Codes: C6130M (Multimedia); C7120 (Financial computing); C7180 (Retailing and distribution computing); C6130E (Data interchange); C7210N (Information networks); C6110J (Object-oriented programming); C6130D (Document processing techniques); C7230 (Publishing and reproduction)

Copyright 2001, IEE

38/5/9 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

07454410 INSPEC Abstract Number: C2000-02-6160M-008

Title: Formal specification of an object -oriented database for intelligent multimedia presentations

Author(s): Huan-Chao Keh; Shih, T.K.; Ching-Sheng Wang

Author Affiliation: Dept. of Comput. Sci. & Inf. Eng., Tamkang Univ., Tamsui, Taiwan

Journal: International Journal of Applied Software Technology vol.4, no.4 p.181-213

Publisher: International Academic Publishing,

Country of Publication: Canada

CODEN: IJSTFX ISSN: 1198-5577

SICI: 1198-5577()4:4L.181:FSOO;1-9

Material Identity Number: D019-1999-001

Language: English Document Type: Journal Paper (JP)

Treatment: Practical (P)

Abstract: The advantages of using a formal specification are in its precise definition of **object** properties and its declarative description of operations of a system. The Z notation is used widely in the literature of formal system designs and the research of language semantics. In line with the rapid growth of multimedia computation research, we use the Z notation to describe a multimedia database that supports the reuse of multimedia presentations. The database consists of two layers: the frame layer and the resource layer. A frame group in the first layer is the basic presentation unit to be reused. A resource group in the second layer is a collection of resources to be used by a frame, or a frame group. In the hierarchy, we use four types of **links** to group **frames** and/or resources.

The main contribution of this paper is in its reuse mechanism of multimedia **objects** in a multimedia database. To show the result, our two multimedia presentation design and generation systems are described. (34 Refs)

Subfile: C

Descriptors: formal specification; multimedia databases; **object** -oriented databases; specification languages

Identifiers: formal specification; **object** -oriented database; intelligent multimedia presentations; Z notation; formal system designs; language semantics; multimedia database; frame layer; resource layer;

software reuse

Class Codes: C6160M (Multimedia databases); C6160J (Object-oriented databases); C6110F (Formal methods)

Copyright 2000, IEE

38/5/11 (Item 4 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

05895262 INSPEC Abstract Number: C9504-6110B-044

Title: Reuse in practice: an industrial perspective

Author(s): Goldberg, M.

Author Affiliation: Rational Software Corp., Santa Clara, CA, USA

Journal: Object Magazine vol.4, no.8 p.46-8

Publication Date: Jan. 1995 Country of Publication: USA

ISSN: 1055-3614

Language: English Document Type: Journal Paper (JP)

Treatment: Applications (A)

Abstract: Effective reuse requires management planning, team coordination, production quality tools, and **modern software** engineering

technology. The article offers a snapshot of industrial practice in **software reuse** from three projects. We look at **object** technology as it relates to reuse, and gain a perspective on how companies, both in the US and overseas, are managing to apply **modern software engineering principles** to real-life **applications**. The **applications** are: shipboard command, control, and communications system; **Windows** -based banking **application**; and enterprise-wide information database. (1 Refs)

Subfile: C

Descriptors: bank data processing; business data processing; command and control systems; **object** -oriented databases; **object** -oriented programming; ships; **software reusability**

Identifiers: industrial perspective; management planning; team coordination; production quality tools; **modern software engineering technology**; industrial practice; **software reuse**; **object** technology; US; **software engineering principles**; real-life **applications**; shipboard command; **Windows** -based banking **application**; enterprise-wide information database

Class Codes: C6110B (Software engineering techniques); C6110J (Object-oriented programming); C6160J (Object-oriented databases); C7120 (Financial computing); C7420 (Control engineering computing)

Copyright 1995, IEE

38/5/13 (Item 6 from file: 2)

DIALOG(R) File 2:INSPEC

(c) 2006 Institution of Electrical Engineers. All rts. reserv.

05349538 INSPEC Abstract Number: B9304-0140B-003, C9304-6150J-001

Title: Operating systems spread 32 -bit wings

Author(s): Van Tyle, S.

Journal: Electronic Design vol.41, no.1 p.93-5

Publication Date: 7 Jan. 1993 Country of Publication: USA

CODEN: ELODAW ISSN: 0013-4872

U.S. Copyright Clearance Center Code: 0013-4872/93/\$1.00+.50

Language: English Document Type: Journal Paper (JP)

Treatment: General, Review (G)

Abstract: The latest wave of systems **software** not only capitalizes on powerful processors, but addresses the increasing importance of multimedia. In 1993, PC **application** developers can look for a much wider range of platforms to work on and greater flexibility within operating systems. Developers also will face more complexity in creating **applications** because of such loose threads as lack of standards and rivalry between the Windows and OS/2 development camps. For now, **application** developers will be able to work with more and more **object** orientation. They'll be able to take advantage of such features as 32-bit instruction and data addressing, and flat (unsegmented) memory models that support **programs** with up to 512 Mbytes of address space. Several factors are driving trends in advanced operating systems for PCs. One is recognition that once information is converted to digital form, all of it-whether text, **graphics**, audio, or video-can be combined on **screen** and linked dynamically. As a result, look for advanced operating systems to better handle multimedia's large data **objects**. Advanced operating systems must be able to run large **applications** while they allow enhanced file-system functions and integration of **applications** based on **object** -oriented technology. Indeed, **object** -oriented programming (OOP) will be key in the coming decade to ease **software** development. (0 Refs)

Subfile: B C

Descriptors: microcomputers; multimedia systems; **object** -oriented programming; operating systems (computers); personal computing; technological forecasting

Identifiers: **32** -bit data addressing; flat memory models; unsegmented memory; powerful processors; multimedia; operating systems; lack of standards; rivalry; Windows; OS/2 development camps; **object** orientation; **32** -bit instruction; address space; **object** -oriented programming; OOP ; **32** bit

Class Codes: B0140B (Planning); C6150J (Operating systems); C6110J (Object-oriented programming); C6160S (Spatial and pictorial databases)

Numerical Indexing: word length 3.2E+01 bit

38/5/14 (Item 7 from file: 2)
DIALOG(R)File 2:INSPEC
(c) 2006 Institution of Electrical Engineers. All rts. reserv.

04905024 INSPEC Abstract Number: C91042141
Title: Visual programming in the ObjectWorld
Author(s): Penz, F.
Author Affiliation: Dept. of Comput. Sci., Vienna Univ., Austria
Journal: Journal of Visual Languages and Computing vol.2, no.1 p. 17-41
Publication Date: March 1991 Country of Publication: UK
ISSN: 1045-926X
U.S. Copyright Clearance Center Code: 1045-926X/91/010014+25\$03.00/0
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The **software** development environment ObjectWorld combines **object** -oriented programming and visual programming to enable **software reuse** . A programmer builds **new objects** by the direct manipulation of prefabricated visual **objects** appearing on the computer **screen** . The ObjectWorld tries to remove some of the problems identified with other visual programming attempts by the consequent **application** of **object** -oriented concepts and other ideas. A sequence of figures explains the concepts of the ObjectWorld. A simple stop-watch is assembled to give a short example of programming in the ObjectWorld. (11 Refs)
Subfile: C
Descriptors: **object** -oriented programming; programming environments; **software reusability** ; visual programming
Identifiers: **software** development environment; ObjectWorld; **object** -oriented programming; visual programming; **software reuse** ; direct manipulation; prefabricated visual **objects** ; stop-watch
Class Codes: C6115 (Programming support); C6130B (Graphics techniques); C6180 (User interfaces)

38/5/16 (Item 1 from file: 94)
DIALOG(R)File 94:JICST-EPlus
(c)2006 Japan Science and Tech Corp(JST). All rts. reserv.

02301656 JICST ACCESSION NUMBER: 95A0205262 FILE SEGMENT: JICST-E
VOC: A visual language aimed to develop applications with graphical user interface.
TASHIRO SHUICHI (1)
(1) Electrotech. Lab., Agency of Ind. Sci. and Technol.
Denshi Gijutsu Sogo Kenkyujo Iho(Bulletin of the Electrotechnical Laboratory), 1994, VOL.58,NO.10, PAGE.901-908, FIG.5, REF.12
JOURNAL NUMBER: F0014ABN ISSN NO: 0366-9092
UNIVERSAL DECIMAL CLASSIFICATION: 681.3.06.004.14:800.92 681.51:007.51
LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan
DOCUMENT TYPE: Journal
ARTICLE TYPE: Original paper

MEDIA TYPE: Printed Publication

ABSTRACT: The use of icons is effective for both **application** 's user interfaces and building **programs** . In this paper, the author proposes a new **object** -oriented visual language system VOC which is both an iconic programming and a user interface designing system. Programmers define and connect icons to construct **applications** . Iconic **programs** and Objective-C upper compatible textual **programs** can be combined to define classes. A new framework named MVCC to improve **software reusability** is also proposed. In the framework, a new factor "Customizer" is added to Smalltalk's Model-View-Controller framework. Each **software** parts, which are mainly for user interfaces like virtual button, slider, etc. can contain Customizer. Customizers start to run when icons are about to be instantiated from classes, and modify the specifications of the **objects** . Programmers can create arbitrary **software** parts without restriction from given customizing tools.
(author abst.)

DESCRIPTORS: **object** -oriented language; user interface; visualization; support **program** ; computation model; computer system development; **window** system; system description language; visual language; graphical user interface

BROADER DESCRIPTORS: programming language; formal language; language; interface; modification; computer **program** ; **software** ; model; development; method

CLASSIFICATION CODE(S): JD03052Y; IB03000G

38/5/17 (Item 1 from file: 6)

DIALOG(R)File 6:NTIS

(c) 2006 NTIS, Intl Cpyrght All Rights Res. All rts. reserv.

1736043 NTIS Accession Number: AD-M000 037/2

**Reuseable Graphical Browser
Software)**

Electronic Systems Div., Hanscom AFB, MA.

Corp. Source Codes: 054864000

Report No.: DOD/SW/MT-93/061

25 Jan 91 mag tape

Languages: English

Journal Announcement: GRAI9317

System: UNIX/SUN3; TAR operating system. Language: Ada and C.

Available in 9-track, ASCII character set tape, 6250 bpi. Documentation included; may be ordered separately as AD-A234445. Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.

NTIS Prices: CP T02

Country of Publication: United States

The **reusable software** component implements a graphical user interface for browsing over the contents of an **object** management system. It is intended to facilitate the construction of various browsing tools by serving as a user interface component for those tools. The generic **application** interface and tailorable user interface provided by the component allow it to be adapted to a wide range of browsing **applications** . They also serve to promote the portability of graphical browsing tools by insulating them from the details of the underlying graphics system. Because the details of the graphics system are hidden by an abstract **application** interface, tool builders need not be familiar with the intricacies of graphics packages for **window** system in **order** to use the product. The RGB is implemented in Ada, with the exception of one small routine coded in C. The current implementation uses the MIT X Toolkit: Ada Language

Implementation (based on X Version 11, Release 3) as the underlying graphics system. This makes it compatible with numerous hardware platforms.

Descriptors: ***Software** ; *Computer graphics; *Man computer interface;
Software tools; **Applications programs** (Computers); Magnetic tapes
Identifiers: *Browsing; **Software reuse** ; Ada programming language; C programming language; NTISDODAF

Section Headings: 62B (Computers, Control, and Information Theory--Computer Software)

38/5/19 (Item 1 from file: 99)

DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs
(c) 2006 The HW Wilson Co. All rts. reserv.

1270810 H.W. WILSON RECORD NUMBER: BAST95066602

OO -VHDL: object -oriented extensions to VHDL

Swamy, Sowmitri; Molin, Arthur; Covnot, Burt

Computer v. 28 (Oct. '95) p. 18-26

DOCUMENT TYPE: Feature Article ISSN: 0018-9162 LANGUAGE: English

RECORD STATUS: New record

ABSTRACT: **Object** -oriented approaches to **software** development have gained widespread acceptance as a way to manage design complexity and increase **software reuse** . These same needs are driving efforts to add **object** -oriented capabilities to hardware description languages (HDLs). VHDL (VHSIC hardware description language) was developed under the auspices of the US Department of Defense's Very High Speed Integrated Circuits (VHSIC) **program** in the 1980s. The **current** DoD effort, called RASSP (Rapid Prototyping of **Application** -Specific Signal Processors), is an ambitious undertaking to reduce, by a factor of four, the cost and time needed to design, upgrade, and replace embedded digital signal processors. To reach this goal, VHDL and its extensions must be used throughout the design process and at all levels from system to gate. VHDL has a number of constructs that have direct correlations in digital hardware; the most important are the component, which encapsulates a "black box " view of a piece of hardware, and the signal, which models a wire. Such constructs make VHDL suitable for developing detailed low-level models, but they make it difficult to write abstract high-level models. For system-level descriptions of behavior, a computation model more in line with **software** systems seems suitable. In this model, individual modules are reactive; that is, they respond only to a command/instruction or stimulus, communication is point-to-point, and the individual addressability of modules allows for temporary implicit communication pathways without the specificity of ports and interconnections. OO -VHDL, the **object** -oriented extension language of VHDL described in this article, supports both the VHDL computation model and the reactive computation model. The authors have implemented a preprocessor that translates OO -VHDL to VHDL and a debugging tool that maps VHDL statements into the OO -VHDL statements from which they were derived. Thus, modelers will be able to use OO -VHDL in current VHDL environments. Copyright 1995, IEEE.

DESCRIPTORS: **Object** -oriented programming; VHDL (Computer hardware description language); Prototyping (Computer science;

38/5/20 (Item 2 from file: 99)

DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs
(c) 2006 The HW Wilson Co. All rts. reserv.

1207798 H.W. WILSON RECORD NUMBER: BAST95003172

The Oberon/F system

Pountain, Dick;

Byte v. 20 (Jan. '95) p. 227-8

DOCUMENT TYPE: Feature Article ISSN: 0360-5280 LANGUAGE: English

RECORD STATUS: New record

ABSTRACT: Oberon/F (commercial version, \$350; educational version, free) is the newest **object** -oriented component framework, a **new software** category. Under development by Niklaus Wirth's Oberon Microsystems, Oberon/F offers a thin **object** -services layer that runs on top of the host operating system and permits the writing of cross-platform, portable, and extensible **applications** as if the host operating system supported full **object** orientation. It is a fully document-centric framework in which everything is a document that can be edited within the development system, which also acts as the runtime system. It incorporates a highly effective but proprietary compound document model that allows for views to be embedded into each other in arbitrarily complex ways. Currently in beta for **Windows** and Macintosh System 7, Oberon/F versions are also planned for OS/2 and Unix/Motif. In future releases, it will be progressively integrated with OLE and OpenDoc.

DESCRIPTORS: Oberon (Computer language); **Reusable software** components;
?

File 347:JAPIO Nov 1976-2005/Oct(updated 060203)

(c) 2006 JPO & JAPIO

File 350:Derwent WPIX 1963-2006/UD,UM &UP=200614

(c) 2006 Thomson Derwent

Set	Items	Description
S1	1677155	PROGRAM? ? OR SOFTWARE OR APPLICATION? ? OR APP OR APPS OR CODE
S2	2286698	PANEL? ? OR PANE OR PANES OR WINDOW? ? OR SCREEN? ? OR BOX OR BOXES OR FRAME OR FRAMES
S3	233821	(SERIES OR SEQUENCE? ? OR SUCCESSIV? OR SUCCESSION OR SUCCEEDING OR SUBSEQUENT?? OR TRANSITION? OR ORDER??? OR CONSECUTIV? OR PROGRESSION OR CHAIN??? OR TRAIN? ? OR LINK??? OR CONNECT???) (5N)S2
S4	6820	(NEXT OR ENSUING OR FOLLOWING) (3W)S2
S5	1123	S2(7N)(GRAPH? ? OR GRAPHED OR GRAPHING OR DIGRAPH?)
S6	31842	(INVOK??? OR INVOCATION OR CALL???) (5N)(METHOD? ? OR PROCEDURE? ? OR FUNCTION? ?)
S7	540234	OBJECT? ? OR OO OR OOP OR OOPPL OR OOPLA OR OLE
S8	1398	S1(5N)(LEGACY OR OLD OR OLDER OR OUTDATED OR OUT(1W)DATE OR PREEXIST? OR PRE()EXIST?)
S9	31790	S1(5N)(MODERN OR NEW OR UPDAT? OR CURRENT OR RECENT OR BACKWARD()COMPAT?)
S10	2805	(S1 OR OBJECT? ? OR COMPONENT? ?) (5N)(REUSE? ? OR REUSING - OR REUSABLE OR REUSABILITY OR RE() (USE OR USING OR USABLE OR - USABILITY))
S11	16096	S1 AND S3:S5
S12	994	S11 AND S6:S7
S13	18	S12 AND S8:S10
S14	11678	S1 AND S2 AND S6:S7
S15	201	S14 AND S8:S10
S16	201	S13 OR S15
S17	22	S16 AND AC=US/PR AND AY=(1963:1995)/PR
S18	25	S16 AND AC=US AND AY=1963:1995
S19	25	S16 AND AC=US AND AY=(1963:1995)/PR
S20	49	S16 AND PY=1963:1995
S21	60	S17:S20
S22	60	IDPAT (sorted in duplicate/non-duplicate order)
S23	11100	(SERIES OR SEQUENCE? ? OR SUCCESSIV? OR SUCCESSION OR SUCCEEDING OR CONSECUTIV? OR PROGRESSION) (3W)S2
S24	155526	(TRANSITION? OR ORDER??? OR CHAIN??? OR TRAIN? ? OR LINK??? OR CONNECT???) (3N)S2
S25	11	S1 AND S23:S24 AND S6 AND S7
S26	35	S1 AND S23:S24 AND S6
S27	542	S1 AND S23:S24 AND S7
S28	306	S27 AND (METHOD? ? OR PROCEDURE? ? OR FUNCTION? ?)
S29	56106	S7(10N)(METHOD? ? OR PROCEDURE? ? OR FUNCTION? ?)
S30	107	S27 AND S29
S31	133	S25:S26 OR S30
S32	128	S31 NOT S16
S33	11	S32 AND AC=US/PR AND AY=(1963:1995)/PR
S34	18	S32 AND AC=US AND AY=1963:1995
S35	18	S32 AND AC=US AND AY=(1963:1995)/PR
S36	20	S32 AND PY=1963:1995
S37	26	S33:S36

22/5/5 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

011880402 **Image available**

WPI Acc No: 1998-297312/199826

Related WPI Acc No: 1996-476666; 1998-609850

XRPX Acc No: N98-232656

Database interface e.g. GUI for application development in client or server environment - includes multiple visual presentation attributes that determines display manner of data window for display of data elements of database table.

Patent Assignee: POWERSOFT CORP (POWE-N)

Inventor: SHEFFIELD K A

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5752018	A	19980512	US 91747858	A	19910820	199826 B
			US 95393049	A	19950223	
			US 96652731	A	19960522	

Priority Applications (No Type Date): US 91747858 A 19910820; US 95393049 A 19950223; US 96652731 A 19960522

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5752018	A		39	G06F-017/30	Cont of application US 91747858 Cont of application US 95393049 Cont of patent US 5566330

Abstract (Basic): US 5752018 A

The interface includes a data **window object** (24) tailored to database table without access to source **code**. Multiple visual presentation attributes are provided which determines the display manner of the data **window** for display of data elements of the database table. The data definition attributes define both type and value validity criteria of data elements.

The data **window object** comprises unit for indirectly manipulating data elements by retrieving data elements from **application** database table into a memory buffer and presenting them in a defined display manner, receiving from a user of interface comments for altering the retrieval data elements in memory buffer and generating database statements automatically to apply results of indirect manipulation from data elements in memory buffer to **application** database table.

ADVANTAGE - Suits **application** programmer who does not have extensive knowledge of database operations or **code** database operations in programming languages. Enables using as modules in numerous **application programs** without reprogramming. Enables **reuse** of previously defined data **windows** thereby decreasing development and testing time.

Dwg.8/30

Title Terms: DATABASE; INTERFACE; APPLY; DEVELOP; CLIENT; SERVE; ENVIRONMENT; MULTIPLE; VISUAL; PRESENT; ATTRIBUTE; DETERMINE; DISPLAY; MANNER; DATA; **WINDOW**; DISPLAY; DATA; ELEMENT; DATABASE; TABLE

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

22/5/8 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

010968239 **Image available**

WPI Acc No: 1996-465188/199646

Related WPI Acc No: 1999-383850

XRPX Acc No: N96-391674

Application development system for computer programme - forms hierarchies based upon elements and behaviours created by authors and uses these as environmental frame of reference

Patent Assignee: MFACTORY INC (MFAC-N); QUARK MEDIA HOUSE BV (QUAR-N)

Inventor: FORSYTHE R H; GUDMUNDSON N K; LEE W A

Number of Countries: 071 Number of Patents: 007

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9631822	A1	19961010	WO 96US4496	A	19960401	199646	B
AU 9654397	A	19961023	AU 9654397	A	19960401	199707	
US 5680619	A	19971021	US 95415848	A	19950403	199748	
EP 826173	A1	19980304	EP 96911536	A	19960401	199813	
			WO 96US4496	A	19960401		
JP 2000505919	W	20000516	JP 96530422	A	19960401	200032	
			WO 96US4496	A	19960401		
EP 826173	B1	20030102	EP 96911536	A	19960401	200310	
			WO 96US4496	A	19960401		
DE 69625592	E	20030206	DE 625592	A	19960401	200318	
			EP 96911536	A	19960401		
			WO 96US4496	A	19960401		

Priority Applications (No Type Date): US 95415848 A 19950403

Cited Patents: 2.Jnl.Ref; US 5493680

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 9631822 A1 E 201 G06F-009/40

Designated States (National): AL AM AT AU AZ BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE HU IS JP KE KG KP KR KZ LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK TJ TM TR TT UA UG US UZ VN

Designated States (Regional): AT BE CH DE DK EA ES FI FR GB GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG

AU 9654397 A G06F-009/40 Based on patent WO 9631822

US 5680619 A 88 G06F-009/40

EP 826173 A1 E G06F-009/40 Based on patent WO 9631822

Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

JP 2000505919 W 211 G06F-009/44 Based on patent WO 9631822

EP 826173 B1 E G06F-009/40 Based on patent WO 9631822

Designated States (Regional): AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

DE 69625592 E G06F-009/40 Based on patent EP 826173

Based on patent WO 9631822

Abstract (Basic): WO 9631822 A

The **application** development system includes one class of element **objects** from which one or more elements can be formed. The class defines a set of characteristics inherent to each **object**. A second class of modifier **objects** allows modifiers to be formed. This defines a second set of characteristics inherent to each modifier. A hierarchical linking mechanism joins an element to a modifier.

A second hierarchical linking mechanism allows an author to link one element as a parent to a second element from the first class. The first element provides a **frame** of reference for the second element and a first of the modifiers.

USE/ADVANTAGE - For interactive multimedia. Allows **objects** to be **reused** in different environments. Allows more complex **applications** to be developed.

Dwg.1/35

Title Terms: APPLY; DEVELOP; SYSTEM; COMPUTER; PROGRAMME; FORM; BASED; ELEMENT; ENVIRONMENT; **FRAME**; REFERENCE

Derwent Class: T01

International Patent Class (Main): G06F-009/40; G06F-009/44

File Segment: EPI

22/5/9 (Item 9 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

010824607 **Image available**

WPI Acc No: 1996-321560/199632

XRPX Acc No: N96-270668

Object -oriented operating system for managing user interface objects
in windows -oriented GUI - in which user interface objects stored in
archive are created via separate constructor program to edit standard
user interface objects and create new user interface objects

Patent Assignee: OBJECT TECHNOLOGY LICENSING CORP (OBJE-N); TALIGENT INC
(TALI-N)

Inventor: PALEVICH J H

Number of Countries: 020 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9615494	A1	19960523	WO 95US13828	A	19951026	199632 B
US 5652884	A	19970729	US 94339112	A	19941114	199736

Priority Applications (No Type Date): US 94339112 A 19941114

Cited Patents: 3.Jnl.Ref; EP 304072; EP 438877; EP 587394; WO 9415284

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

WO 9615494	A1	E	88		
------------	----	---	----	--	--

Designated States (National): CA CN JP

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL
PT SE

US 5652884	A		56		
------------	---	--	----	--	--

Abstract (Basic): WO 9615494 A

The **object** oriented operating system includes user interface
objects which are stored in a user interface **object** archive which is
a database physically located in the shared library of an associated
application program. The **objects** are stored in a hierarchical
locale tree within the archive, in order to enable preparation of an
application developed in one language for use in an 'area' or locale
which uses another language. As the tree proceeds away from the root
locale, the locales become more specific as to language. All **objects**
are stored in the base or root of the hierarchy, but only **objects**
which require a translation are stored in an area associated with a
more specific locale.

At run-time, a complete collection of **objects** is assembled by
starting at the desired locale and proceeding up the hierarchy
level-by-level. Translated **objects** at lower levels of the hierarchy
'override' those at higher levels such that the most complete
translations of each **objects** are obtained during the search.

USE/ADVANTAGE - Provides user interface **object** archiving system
which can manage user **objects** to ensure consistency between various
sections of projects which use common **objects**. Can easily accommodate
newly-designed user interface **objects** without requiring rewrite of
object editor.

Dwg.1/35

Title Terms: **OBJECT** ; ORIENT; OPERATE; SYSTEM; MANAGE; USER; INTERFACE;
OBJECT ; **WINDOW** ; ORIENT; USER; INTERFACE; **OBJECT** ; STORAGE; ARCHIVE;
SEPARATE; CONSTRUCTION; **PROGRAM** ; EDIT; STANDARD; USER; INTERFACE;
OBJECT ; NEW; USER; INTERFACE; **OBJECT**

Derwent Class: T01

International Patent Class (Main): G06F-009/44; G06F-017/30

File Segment: EPI

22/5/10 (Item 10 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

010824605 ****Image available****

WPI Acc No: 1996-321558/199632

XRPX Acc No: N96-270666

Object -oriented operating system for managing user interface objects in windows -oriented GUI - has parts palette which provides framework for users of interface builder, to collect, categorise and reuse UI objects

Patent Assignee: TALIGENT INC (TALI-N)

Inventor: LENZ M A

Number of Countries: 017 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9615492	A1	19960523	WO 95US13753	A	19951026	199632 B

Priority Applications (No Type Date): US 94340085 A 19941114

Cited Patents: 1.Jnl.Ref; WO 9415284

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9615492	A1	E	94	G06F-009/44	

Designated States (National): CA

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE

Abstract (Basic): WO 9615492 A

The **object** oriented operating system includes user interface **objects** which are stored in a user interface **object** archive which is a database physically located in the shared library of an associated **application program**. The **objects** are stored in a hierarchical locale tree within the archive, in order to enable preparation of an **application** developed in one language for use in an 'area' or locale which uses another language. At run-time, a complete collection of **objects** is assembled by starting at the desired locale and proceeding up the hierarchy level-by-level. Translated **objects** at lower levels of the hierarchy 'override' those at higher levels such that the most complete translations of each **objects** are obtained during the search.

The user interface **objects** stored in the archive are created using a predefined constructor **program**, and in order to allow newly-created user interface **objects** to use the constructor **program**, each user interface is contained in a special escort **object** that interfaces with the constructor **program**. The parts palette provides a framework for users of constructor, a user interface builder, to

collect, categorise and **reuse** UI **objects**. Users can add custom **objects** e.g controls, views and dialogues to the parts palette, group them in any number of **panels**, and **reuse** **objects** by dragging and dropping them into their own **applications**.

USE/ADVANTAGE - Managing user **objects** to ensure consistency between various sections of projects which use common **objects** in user interface **object** archiving system. Easily accommodates newly-designed user interface **objects** without rewriting of **object** editor.

Dwg.1/35

Title Terms: **OBJECT**; ORIENT; OPERATE; SYSTEM; MANAGE; USER; INTERFACE;
OBJECT; **WINDOW**; ORIENT; PART; PALLET; FRAMEWORK; USER; INTERFACE;
BUILD; COLLECT; CATEGORY; REUSE; UI; **OBJECT**

Derwent Class: T01

International Patent Class (Main): G06F-009/44

File Segment: EPI

22/5/11 (Item 11 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

010771237 **Image available**

WPI ACC No: 1996-268191/199627

XRPX ACC No: N96-225500

Link manager for managing links integrating data between application programs - in which container objects contain references to linked data residing in server objects , and uses link mechanism for invoking server application program to provide linked data to presentation window

Patent Assignee: KODAK LTD (EAST); WANG LAB INC (WANG)

Inventor: PRATT J M

Number of Countries: 020 Number of Patents: 008

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9609585	A1	19960328	WO 95US9890	A	19950804	199627 B
AU 9532125	A	19960409	AU 9532125	A	19950804	199629
EP 729610	A1	19960904	EP 95928308	A	19950804	199640
			WO 95US9890	A	19950804	
JP 9506457	W	19970624	WO 95US9890	A	19950804	199735
			JP 96510872	A	19950804	
US 5778385	A	19980707	US 94309824	A	19940921	199834
			US 97891240	A	19970710	
AU 707240	B	19990708	AU 9532125	A	19950804	199938
EP 729610	B1	20000712	EP 95928308	A	19950804	200036
			WO 95US9890	A	19950804	
DE 69517945	E	20000817	DE 617945	A	19950804	200047
			EP 95928308	A	19950804	
			WO 95US9890	A	19950804	

Priority Applications (No Type Date): US 94309824 A 19940921; US 97891240 A 19970710

Cited Patents: EP 369961; EP 500262; EP 578207

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
WO 9609585	A1	E 26	G06F-009/46	
			Designated States (National): AU CA JP	
			Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE	
AU 9532125	A		G06F-009/46	Based on patent WO 9609585
EP 729610	A1	E 26	G06F-009/46	Based on patent WO 9609585
			Designated States (Regional): DE FR GB	
JP 9506457	W	38	G06F-009/46	Based on patent WO 9609585
US 5778385	A		G06F-009/40	Cont of application US 94309824
AU 707240	B		G06F-009/46	Previous Publ. patent AU 9532125
				Based on patent WO 9609585
EP 729610	B1	E	G06F-009/46	Based on patent WO 9609585
			Designated States (Regional): DE FR GB	
DE 69517945	E		G06F-009/46	Based on patent EP 729610
				Based on patent WO 9609585

Abstract (Basic): WO 9609585 A

The link manager includes a reference selector for selecting a single reference to linked data in a container **object** , and a moniker generator to generate a moniker identifying a server **object** containing the linked data. The link mechanism responds to the moniker pointer by invoking a server **application program** to open the server **object** and provide the **linked** data to the presentation **window** . Each reference includes a file name and a path name, and a search mechanism determines the server **object** directory path, which is used in generating the moniker.

The link manager responds to a second reference by modifying the moniker to identify a second server **object** , and the link mechanism **reuses** the link by closing the first **application program** and server **object** , and invoking a second server **application program** to open the second server **object** , and provide the **linked** data to

the presentation **window** . A moniker manager allows the opening of several links, each with a moniker, but directs the moniker generator to modify and reuse a previous moniker and link when at a preselected limit.

USE/ADVANTAGE - Integrating data between **applications** **programs** executing in integrated operating environment

Dwg.1/2a

Title Terms: LINK; MANAGE; MANAGE; LINK; INTEGRATE; DATA; APPLY; **PROGRAM** ; CONTAINER; **OBJECT** ; CONTAIN; REFERENCE; LINK; DATA; SERVE; **OBJECT** ; LINK; MECHANISM; INVOKE; SERVE; APPLY; **PROGRAM** ; LINK; DATA; PRESENT; **WINDOW**

Derwent Class: T01

International Patent Class (Main): G06F-009/40; G06F-009/46

International Patent Class (Additional): G06F-009/06

File Segment: EPI

22/5/13 (Item 13 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

010733822 **Image available**

WPI Acc No: 1996-230777/199623

XRPX Acc No: N96-193717

Object oriented system for servicing windows - has window server for creating window objects in response to receipt of parameters from one or more clients and ordering window areas according to type specified by parameter

Patent Assignee: OTL CORP (OTLO-N); TALIGENT INC (TALI-N); OBJECT TECHNOLOGY LICENSING CORP (OBJE-N)

Inventor: LYNCH-FRESHNER L A; MARSH D M; MILNE S H; ZIAS J A

Number of Countries: 020 Number of Patents: 006

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9613026	A1	19960502	WO 95US10632	A	19950817	199623 B
EP 788646	A1	19970813	EP 95930882	A	19950817	199737
			WO 95US10632	A	19950817	
US 5668997	A	19970916	US 94328230	A	19941025	199743
			US 95573916	A	19951218	
JP 10507853	W	19980728	WO 95US10632	A	19950817	199840
			JP 96513878	A	19950817	
EP 788646	B1	19990428	EP 95930882	A	19950817	199921
			WO 95US10632	A	19950817	
DE 69509406	E	19990602	DE 609406	A	19950817	199928
			EP 95930882	A	19950817	
			WO 95US10632	A	19950817	

Priority Applications (No Type Date): US 94328230 A 19941025; US 95573916 A 19951218

Cited Patents: 10Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

WO 9613026	A1	E	46	G09G-005/14	
------------	----	---	----	-------------	--

Designated States (National): CA CN JP

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE

EP 788646	A1	E		G09G-005/14	Based on patent WO 9613026
-----------	----	---	--	-------------	----------------------------

Designated States (Regional): DE FR GB

US 5668997	A		31	G06F-003/14	Cont of application US 94328230
------------	---	--	----	-------------	---------------------------------

JP 10507853	W		72	G06F-003/14	Based on patent WO 9613026
-------------	---	--	----	-------------	----------------------------

EP 788646	B1	E		G09G-005/14	Based on patent WO 9613026
-----------	----	---	--	-------------	----------------------------

Designated States (Regional): DE FR GB

DE 69509406	E			G09G-005/14	Based on patent EP 788646
-------------	---	--	--	-------------	---------------------------

Based on patent WO 9613026

Abstract (Basic): WO 9613026 A

The **window** server system communicates with clients and creates, deletes and modifies **window objects**. **Objects** are created in response to parameters provided by clients, who can obtain a variety of information regarding **windows** managed by the **window** server. Hardware **windows** are supported by sub-classing **objects** which provide polymorphic **screen objects**, such that it does not matter whether the **window** is created by a **software** or hardware entity.

Clients may be notified by the **window** server in response to certain events occurring w.r.t particular **windows**, e.g a configuration change. The **window** server also dynamically manages a default **window** layering scheme which takes into account the parameters specified, or not specified, by clients as well as the characteristics of the **windows** currently being managed by the **window** server.

USE/ADVANTAGE - Managing **window** display areas in graphical user interface. Enables interfacing with **application programs** such that **screen** display can be rapidly modified. Coordinates display generation for all **application programs** to prevent **applications** or clients interfering with each other or overwriting each other on **screen** display.

Dwg.3/22

Title Terms: **OBJECT**; ORIENT; SYSTEM; SERVICE; **WINDOW**; **WINDOW**; SERVE; **WINDOW**; **OBJECT**; RESPOND; RECEIPT; PARAMETER; ONE; MORE; CLIENT; ORDER; **WINDOW**; AREA; ACCORD; TYPE; SPECIFIED; PARAMETER

Derwent Class: P85; T01

International Patent Class (Main): G06F-003/14; G09G-005/14

International Patent Class (Additional): G06F-009/44; G09G-005/00

File Segment: EPI; EngPI

22/5/20 (Item 20 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

010221037 **Image available**

WPI Acc No: 1995-122292/ 199516

XRPX Acc No: N95-096670

Dynamic interface builder for simulating industry - is based on parameter changes of application system to real-time change displayed screen according to user requirement using multiple dynamic object generators

Patent Assignee: INST INFORMATION IND (INFO-N)

Inventor: JANG C; SHEEN J; SHIAU Y

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
TW 239883	A	19950201	TW 93103502	A	19930504	199516 B

Priority Applications (No Type Date): TW 93103502 A 19930504

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
TW 239883	A		21	G06F-015/21	

Abstract (Basic): TW 239883 A

The dynamic interface builder includes a view generator which generates one or more views on the **window**. The content of view is dependent on the group contents of the generator. A group generator defines group **object** by recursive method. A new group generator redefines the dynamic parameters of new group by using one known group **object** to make the old group **reusable**. A basic graphic set storage device stores every basic graphic shape such as circle, rectangle, line segment, arc, ellipse etc., coupled with the group generator to supply the needed graphic to the group generator.

An interactive set generator supplies the interface signal to the user, coupled with peripheral input equipment such as the keyboard,

mouse, wireless button and main text editor to generate appropriate action corresponding to the input signal. A display set storage device stores the defined dynamic displays, coupled with the group generator to supply the needed dynamic display signal to the group generator based on its requirement. An output dynamic **object** generator with one built-in attribute generator makes the displayed **screen** with different display characteristics.

Dwg.2/5

Title Terms: DYNAMIC; INTERFACE; BUILD; SIMULATE; INDUSTRIAL; BASED; PARAMETER; CHANGE; APPLY; SYSTEM; REAL-TIME; CHANGE; DISPLAY; **SCREEN** ; ACCORD; USER; REQUIRE; MULTIPLE; DYNAMIC; **OBJECT** ; GENERATOR

Derwent Class: T01

International Patent Class (Main): G06F-015/21

File Segment: EPI

22/5/21 (Item 21 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

010098908 **Image available**

WPI ACC No: 1995-000161/ **199501**

XRPX ACC No: N95-000130

Calling **32-bit** functions from **16-bit** functions - building parameter list in stack compatible with **32-bit** calling, translating **16 bit** register values in stack into format accessible to **32-bit** code

Patent Assignee: IBM CANADA LTD (IBMC); INT BUSINESS MACHINES CORP (IBMC)

Inventor: MOONEY D M; STOODLEY K A

Number of Countries: 002 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
CA 2093451	A	19941007	CA 2093451	A	19930406	199501 B
US 5490256	A	19960206	US 9343453	A	19930406	199612 N
			US 94301221	A	19940906	
CA 2093451	C	20000314	CA 2093451	A	19930406	200032

Priority Applications (No Type Date): CA 2093451 A 19930406; US 94301221 A 19940906

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
CA 2093451	A		17	G06F-009/06	
US 5490256	A		7	G06F-009/40	Cont of application US 9343453
CA 2093451	C	E		G06F-009/06	

Abstract (Basic): CA 2093451 A

The method involves building a parameter list in a stack compatible with **32-bit code** , translating **16-bit** register values in the stack into a form accessible to **32-bit code** , changing operating mode in association with the stack to **32-bit** form, and establishing all additional **32-bit** segment registers. Transfer of control is implemented to **32-bit code** .

The method further involves creating and maintaining a record of the original register values for the at least one **16-bit** stack **frame** before performing the steps of building the parameter list and translating said **16-bit** register values.

USE/ADVANTAGE - For hardware implementing of mechanism for converting **16-bit older applications** to **32-bit code** environment. Provision for automatic recognition that **32-bit code** has been called by **16-bit** subroutine and automatic actuation of compiler, rendering different format of **16-bit code** transparent in **32-bit code** environment.

Dwg.1/3

Title Terms: CALL; BIT; FUNCTION; BIT; FUNCTION; BUILD; PARAMETER; LIST; STACK; COMPATIBLE; BIT; CALL; TRANSLATION; BIT; REGISTER; VALUE; STACK;

FORMAT; ACCESS; BIT; **CODE**
Derwent Class: T01
International Patent Class (Main): G06F-009/06; G06F-009/40
International Patent Class (Additional): G06F-005/00
File Segment: EPI

22/5/22 (Item 22 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

010026114 ****Image available****
WPI Acc No: 1994-293827/ **199436**
XRPX Acc No: N94-231246

Defining, creating or editing graphical user interface panel - involves compiling graphical user interface source code file to form GUI object code file and updating size and location object entries, and dynamically sizing

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: HOFFMANN R P; MALCOLM J W; MONTGOMERY J D; STONE S S
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
US 5347627 A 19940913 US 92864766 A 19920407 199436 B

Priority Applications (No Type Date): US 92864766 A 19920407
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 5347627 A 11 G06F-015/00

Abstract (Basic): US 5347627 A

The method involves opening a graphical user interface source **code** file containing an **object** entry for the user interface **panel** and for each **object** within the **panel**. The **object** entries include the size and location of the **object**.

The graphical user interface source **code** file is edited to form an edited graphical user interface source **code** file, and the size and location source entries of each **object**. The edited graphical user interface source **code** file is compiled. The compiled graphical user interface **object code** file is stored in a **Panel** Binary Intermediate file.

ADVANTAGE - Provides end-user with ability to save edited source **code** corresp. to edited GUI **panel**. Enhances ability of end-user to further modify or edit or even copy, GUI file.

Dwg.3/5

Title Terms: DEFINE; EDIT; GRAPHICAL; USER; INTERFACE; **PANEL**; COMPILE; GRAPHICAL; USER; INTERFACE; SOURCE; **CODE**; FILE; FORM; **OBJECT**; **CODE**; FILE; UPDATE; SIZE; LOCATE; **OBJECT**; ENTER; DYNAMIC; SIZE

Derwent Class: T01
International Patent Class (Main): G06F-015/00
International Patent Class (Additional): G06F-003/14
File Segment: EPI

22/5/24 (Item 24 from file: 350)

DIALOG(R)File 350:Derwent WPIX
(c) 2006 Thomson Derwent. All rts. reserv.

009447455
WPI Acc No: 1993-140980/ **199317**
XRPX Acc No: N93-107543

Context-sensitive object icon drag-drop protocol for visual code generation - allows user to drag object from sample pallet and drop it on specific target location in source code window

Patent Assignee: ANONYMOUS (ANON)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
RD 347018	A	19930310	RD 93347018	A	19930220	199317 B

Priority Applications (No Type Date): RD 93347018 A 19930220

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
RD 347018	A		G06F-000/00	

Abstract (Basic): RD 347018 A

When the **object** icon is dragged into a text (or source **code**) **window** for **code** editing, the Source **Code window** is in an edit mode with source **code** lines displayed. The user can move the **object** icon in the Source **Code window** and: If the **current** position is within a **code** line, dropping the icon will replace the **code** in the **code** line by the source **code** represented by the icon. If the **current** position is between two **code** lines, or within an enclosing language expression (e.g.()), dropping the icon will insert the **code** represented by the icon into the corresponding position; if the user intends to replace multiple lines of **code** in the Source **Code window**, the operation will be: dragging the icon to the start of the **code** to be replaced; holding down a designated key (e.g. Shift); moving the icon to the end of the **code** to be replaced; releasing the key and mouse button.

When the icon is dragged into an area designated for visual builder, a visual representation of the sample **object** will be creating. The user may modify the visual **object** under the design mode of this builder and its underlying **code** will change accordingly. When the user drags the visual representation out of the builder area, the visual representation will turn into the visual **object** icon, ready for the next drag-drop operation.

USE/ADVANTAGE - Context-sensitive approach to generating **code** via drag-drop of visual **objects** provided in **object** pallet. When an **object** pallet of sample **objects** is provided, the user can drag on **object** from the pallet and drop it on a specific target location.

Title Terms: SENSITIVE; **OBJECT**; DRAG; DROP; PROTOCOL; VISUAL; **CODE**; GENERATE; ALLOW; USER; DRAG; **OBJECT**; SAMPLE; PALLET; DROP; SPECIFIC; TARGET; LOCATE; SOURCE; **CODE**; **WINDOW**

Derwent Class: T01

International Patent Class (Main): G06F-000/00

File Segment: EPI

22/5/26 (Item 26 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

008800971 **Image available**

WPI ACC No: 1991-304983/ 199142

XRPX ACC No: N91-233633

Interactive computer system for creating new application - links appearance with events to complete programming mode and runs application by interaction interface

Patent Assignee: IBM CORP (IBMC); INT BUSINESS MACHINES CORP (IBMC)

Inventor: HIRAGA R; LIEN Y C; MIMA J; MORISHIMA H; LIEN Y; MIMA Y

Number of Countries: 002 Number of Patents: 005

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 451963	A	19911016	EP 91302183	A	19910314	199142 B
CA 2038265	A	19911001				199151
EP 451963	A3	19920610	EP 91302183	A	19910314	199332
US 5522024	A	19960528	US 91676608	A	19910328	199627
			US 94260509	A	19940615	
			US 95503676	A	19950718	

US 5600780	A	19970204	US 91676608	A	19910328	199711
			US 94260809	A	19940616	
			US 95503676	A	19950718	
			US 96606461	A	19960304	

Priority Applications (No Type Date): JP 9081473 A 19900330

Cited Patents: NoSR.Pub; 2.Jnl.Ref; EP 343882

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5522024	A		13	G06F-003/14	Cont of application US 91676608
					Cont of application US 94260509
US 5600780	A		13	G06F-015/00	Cont of application US 91676608
					Cont of application US 94260809
					Cont of application US 95503676
					Cont of patent US 5522024

Abstract (Basic): EP 451963 A

The system has a programming environment in which an end user of a number of **applications** e.g. integrated environment **applications** can customise them with a unique user interface. A customisation facility records the event stream from the user to the **applications** when they run and enables him to combine several event streams into a 'story'.

The user generates his own unique interface **screen** with graphics e.g. **boxes**, arrows icons etc. using a graphical editor. He then **links** the **screen** to event streams or stories and completes a programming mode.

ADVANTAGE - **Applications** can be customised by non-programmers.

(16pp Dwg.No.1/12)

Title Terms: INTERACT; COMPUTER; SYSTEM; NEW; APPLY; LINK; APPEAR; EVENT; COMPLETE; **PROGRAM**; MODE; RUN; APPLY; INTERACT; INTERFACE

Derwent Class: T01

International Patent Class (Main): G06F-003/14; G06F-015/00

International Patent Class (Additional): G06F-003/02; G06F-009/00;

G06F-009/44

File Segment: EPI

22/5/28 (Item 28 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

007956785 **Image available**

WPI ACC No: 1989-221897/ 198931

XRPX ACC No: N89-169318

Direct cursor-controlled access to multiple application programs - allowing user to invoke function resilient in computer system to directly access windows with chain running programs

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)

Inventor: PETERS A M

Number of Countries: 013 Number of Patents: 007

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 325885	A	19890802	EP 88480099	A	19881220	198931 B
BR 8900302	A	19890919				198943
US 5050105	A	19910917	US 88148730	A	19880126	199140
CA 1306067	C	19920804	CA 582795	A	19881110	199237
EP 325885	B1	19940706	EP 88480099	A	19881220	199426
DE 3850560	G	19940811	DE 3850560	A	19881220	199431
			EP 88480099	A	19881220	
ES 2056951	T3	19941016	EP 88480099	A	19881220	199442

Priority Applications (No Type Date): US 88148730 A 19880126

Cited Patents: 3.Jnl.Ref; A3...9051; EP 273248; No-SR.Pub

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

EP 325885 A E 18
 Designated States (Regional): BE CH DE ES FR GB IT LI NL SE
 EP 325885 B1 E 22 G06F-009/44
 Designated States (Regional): BE CH DE ES FR GB IT LI NL SE
 DE 3850560 G G06F-009/44 Based on patent EP 325885
 ES 2056951 T3 G06F-009/44 Based on patent EP 325885
 CA 1306067 C G06F-009/44

Abstract (Basic): EP 325885 A

Two or more programmes may be optionally linked together in an arbitrary sequence to form a chain. A user may **invoke** a **function** resident in the computer system to directly access **windows** in which programmes in the **chain** are running in a sequence determined by the order in which the user opens the **windows** in a given session. The method for traversing between programmes in the chain involves a uniform, short procedure dependent on positioning of a pointing cursor within an active **window** and, for example, pressing a mouse button.

Further access to data within any **application** programme is available slightly different from that used to navigate between programmes, is still uniform and short when compared with that normally used to traverse **application** programmes.

ADVANTAGE - Sparing in memory requirement.

1/11

Title Terms: DIRECT; CURSOR; CONTROL; ACCESS; MULTIPLE; APPLY; **PROGRAM** ;
 ALLOW; USER; INVOKE; FUNCTION; RESILIENT; COMPUTER; SYSTEM; ACCESS;
WINDOW ; CHAIN; RUN; **PROGRAM**

Derwent Class: P85; T01

International Patent Class (Main): G06F-009/44

International Patent Class (Additional): G06F-003/10; G09G-001/00

File Segment: EPI; EngPI

22/5/51 (Item 51 from file: 347)

DIALOG(R)File 347:JAPIO

(c) 2006 JPO & JAPIO. All rts. reserv.

03920921 **Image available**

OBJECT DISPLAY CONTROL METHOD

PUB. NO.: 04-286021 [JP 4286021 A]
 PUBLISHED: October 12, 1992 (**19921012**)
 INVENTOR(s): KAWAMURA AKIKO

UEDA YOSHIHIRO
 WANISHI MAKOTO
 TSUDA MITSUHIRO
 MIZUGUCHI TAMOTSU
 KAMATA HAJIME

APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP
 (Japan)

APPL. NO.: 03-051337 [JP 9151337]

FILED: March 15, 1991 (19910315)

INTL CLASS: [5] G06F-003/14

JAPIO CLASS: 45.3 (INFORMATION PROCESSING -- Input Output Units)

JOURNAL: Section: P, Section No. 1490, Vol. 17, No. 84, Pg. 165,
 February 19, 1993 (19930219)

ABSTRACT

PURPOSE: To improve the identification of a specified **object** by adding a shadow pattern on the periphery of a display area in the specified **object** among plural **objects** .

CONSTITUTION: The files of the respective **objects** A-C, e.g. are set to be only the files 6-8 of image data for regular display. When the identification flag 10 of an **object** management table 3 designates display with a shadow, a display processing **program** 2 generates and adds the

shadow pattern based on coordinate and size information of the **object** display area. The display processing **program** 2 supervises the **updatation** of the **object** management table 3 by interruption. A processing which immediately switches a display system to a corresponding one when the value of the identification flag 10 is changed is executed. The identification of the specified **object** in a display 5 **screen** where plural **objects** A', B, C are displayed is improved.

22/5/53 (Item 53 from file: 347)

DIALOG(R)File 347:JAPIO

(c) 2006 JPO & JAPIO. All rts. reserv.

03594724 **Image available**

SCREEN LANGUAGE SYSTEM

PUB. NO.: 03-257624 [JP 3257624 A]

PUBLISHED: November 18, 1991 (**19911118**)

INVENTOR(s): AOE SHIGERU
KAKEHI HAJIME
RYU TADAMITSU

APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)

APPL. NO.: 02-057712 [JP 9057712]

FILED: March 08, 1990 (19900308)

INTL CLASS: [5] G06F-009/06; G06F-009/44

JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

JOURNAL: Section: P, Section No. 1312, Vol. 16, No. 60, Pg. 94,
February 14, 1992 (19920214)

ABSTRACT

PURPOSE: To eliminate necessity for a **program** itself when producing a **new** system by separately partitioning a method and an instance in an **object** capsule on a display **screen**, registering them and suitably combining these **object** parts.

CONSTITUTION: An area to store data is dynamically positioned in an instance file 4 as an entity file 3 and the **program** is installed in a method file 5 prepared in advance. Then, an area storing the method is controlled by a table, for example. The index of an **object** sensor is read out from a parts file 2 and displayed on a display **screen** 1 and out of the display, the desirably combined instance **object** sensor and method **object** sensor are selected and registered to an **object** file 6 as the **object** capsule. Then, a **screen** is prepared. Thus, when producing the new system, it is not necessary to prepare the **program** itself.

22/5/54 (Item 54 from file: 347)

DIALOG(R)File 347:JAPIO

(c) 2006 JPO & JAPIO. All rts. reserv.

03567384 **Image available**

METHOD FOR FORMING PROGRAM OBJECT

PUB. NO.: 03-230284 [JP 3230284 A]

PUBLISHED: October 14, 1991 (**19911014**)

INVENTOR(s): TAMURA TOSHIYUKI
ASAI FUMIYASU

APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or Corporation), JP (Japan)

APPL. NO.: 02-026814 [JP 9026814]

FILED: February 05, 1990 (19900205)

INTL CLASS: [5] G06F-015/82; G06F-009/45

JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer **Applications**);
45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units

JOURNAL: Section: P, Section No. 1297, Vol. 16, No. 12, Pg. 6, January 13, 1992 (19920113)

ABSTRACT

PURPOSE: To efficiently use a memory for storing a **program** by applying one identifier(ID) to plural data to be exchanged and executing plural operation processing by one node when plural data to be operated can be mutually exchanged and the operation order of plural operation processing can be exchanged.

CONSTITUTION: In the case of executing arithmetic processing for repeatedly processing logical operation capable of mutually exchanging data such as sum operation processing, multiplying operation processing, OR operation, and AND operation, i.e. in the case of executing operation whose data to be operated can be mutually exchanged and operation order can also be exchanged, the same ID (node number) SUM is applied to the exchangeable data and the arithmetic processing of plural times is executed only by one node SUM. Three nodes using a new instruction code 'SUM' are substituted for a part constituted of nodes #1 to #7 indicated in the shown dotted line **frame**. Consequently, the number of instruction codes to be formed can be reduced, a memory area for storing a **program** can be effectively utilized to suppress the useless consumption of the area and a matching memory can be prevented from being uselessly occupied to smoothly advance the **program**.

37/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2006 JPO & JAPIO. All rts. reserv.

05020153 **Image available**
CODING CONTROL METHOD

PUB. NO.: 07-312753 [JP 7312753 A]
PUBLISHED: November 28, 1995 (19951128)
INVENTOR(s): KUROBE AKIO
APPLICANT(s): MATSUSHITA ELECTRIC IND CO LTD [000582] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 06-104007 [JP 94104007]
FILED: May 18, 1994 (19940518)
INTL CLASS: [6] H04N-007/32; G06T-009/00; H03M-007/30
JAPIO CLASS: 44.6 (COMMUNICATION -- Television); 42.4 (ELECTRONICS -- Basic Circuits); 45.9 (INFORMATION PROCESSING -- Other)

ABSTRACT

PURPOSE: To absorb timewise and spatial distortion by correcting the optimum operating point of an image coder based on a calculated coding rate so as to obtain an optimum balance in the visual characteristic.

CONSTITUTION: A coding control section 210 calculates a generated **code** quantity $R(q)$ and an $S/N D(q)$ to calculate a coding rate $Ss(q)=L/(R(q)+LXPXTn-bn)$, where q is quantization accuracy. In this case, the control section 210 receives a transmittal confirmation bit number bn from a transmission control section 5 in the transmission state of a preceding coding frame and uses the $S/N D(q)$ and the coding rate $Ss(q)$ to calculate an image quality trade-off function $Ss=G(Ds)$. Then the control section 210 calculates q , $Ss(q)$ being cross points between the **function** Ss and an **object function** $So=O(Do)$, starts a quantization section 223 to quantize an image signal by one frame. Then the control section 210 starts 1st and 2nd variable length coding sections 224, 225 to **code** quantized image data and calculates a frame jump time Tn up to a **succeeding coded frame** depending on the coding rate Ss and restore to the original operation.

37/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2006 JPO & JAPIO. All rts. reserv.

04792164 **Image available**
SCREEN DISPLAY PROCESSING SYSTEM

PUB. NO.: 07-084764 [JP 7084764 A]
PUBLISHED: March 31, 1995 (19950331)
INVENTOR(s): IGUCHI MUNEKI
APPLICANT(s): TOSHIBA CORP [000307] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 05-188857 [JP 93188857]
FILED: June 30, 1993 (19930630)
INTL CLASS: [6] G06F-009/06
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessors)

ABSTRACT

PURPOSE: To easily and efficiently perform the preparation work and the maintenance work of a screen display **program**.

CONSTITUTION: After a screen display processing part is completely separated as an external module from a user **program** 3, a menu system 4 composed of modules 41 to 46 is constituted and the menu system 4 is started by a main function 31 to be the main routine of the user **program**

3, the control processing of a terminal equipment 2, the screen data transmission and reception processing with the terminal equipment 2, the **screen transition** processing described in external operation definition files 5a to 5c and the data operations of screens a to c by the **calling** of sub- **functions** 32a to 32c within the user **program** 3 are made to be performed in accordance with the control module 46 as the main routine within the system 4.

37/5/4 (Item 4 from file: 347)

DIALOG(R)File 347:JAPIO

(c) 2006 JPO & JAPIO. All rts. reserv.

04322172 **Image available**

SIMPLE LANGUAGE PROGRAMMING SYSTEM

PUB. NO.: 05-313872 [JP 5313872 A]

PUBLISHED: November 26, 1993 (**19931126**)

INVENTOR(s): SUGANO MIKITO

APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or Corporation), JP (Japan)

APPL. NO.: 04-116560 [JP 92116560]

FILED: May 11, 1992 (19920511)

INTL CLASS: [5] G06F-009/06

JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

JOURNAL: Section: P, Section No. 1703, Vol. 18, No. 129, Pg. 147, March 03, 1994 (19940303)

ABSTRACT

PURPOSE: To simplify the preparation of an **application program** by writing various parameters into the icons through a **window** and **connecting** these icons together by means of the window and a mouse.

CONSTITUTION: A window generating part **calls** a **function** icon 14 corresponding to a parameter to a main window 12 from an icon display window 3. The icon 14 is set a desired parameter when a cursor is put in a prescribed range and the switch of a mouse is operated. In such a way, a parameter is cset to each icon. Then, a connection line 15 is drawn between each of icons 14 in the window 12 so that a processing flow is shown. Thus, a logical constitution is decided between each of the **program** modules for a desired **application program**. Thus, the parameters are inputted with use of a visual programming tool and therefore an **application** is easily prepared.

37/5/17 (Item 10 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

009967201 **Image available**

WPI Acc No: 1994-234914/ **199428**

XRPX Acc No: N94-185649

Processor system for executing program containing differing types of object code - has procedure field which differentiates between calling convention and can be switched between modes appropriate to object code

Patent Assignee: APPLE COMPUTER INC (APPY)

Inventor: EIDT E L; JONES B K; LILLICH A W; MOUREY J D; PARENT S R; MOUREY J; TRAUT E P

Number of Countries: 045 Number of Patents: 013

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9415283	A1	19940707	WO 93US12148	A	19931214	199428 B
AU 9458012	A	19940719	AU 9458012	A	19931214	199439
US 5452456	A	19950919	US 92993923	A	19921218	199543

EP 674783	A1	19951004	WO 93US12148	A	19931214	199544
			EP 94903634	A	19931214	
JP 8506195	W	19960702	WO 93US12148	A	19931214	199650
			JP 94515254	A	19931214	
EP 674783	B1	19980923	WO 93US12148	A	19931214	199842
			EP 94903634	A	19931214	
DE 69321255	E	19981029	DE 93621255	A	19931214	199849
			WO 93US12148	A	19931214	
			EP 94903634	A	19931214	
CA 2152041	C	19990316	CA 2152041	A	19931214	199929
US 6256658	B1	20010703	US 92993923	A	19921218	200140
			US 95409477	A	19950322	
JP 2004046881	A	20040212	JP 94515254	A	19931214	200413
			JP 2003277282	A	20030722	
JP 3508775	B2	20040322	WO 93US12148	A	19931214	200421
			JP 94515254	A	19931214	
JP 2005149532	A	20050609	JP 2003277282	A	19931214	200538
			JP 200518957	A	20050126	
JP 3705547	B2	20051012	JP 94515254	A	19931214	200566
			JP 2003277282	A	20030722	

Priority Applications (No Type Date): US 92993923 A 19921218; US 95409477 A 19950322

Cited Patents: EP 387172; EP 414624; US 4084235; US 4812975

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

WO 9415283	A1	E	39	G06F-009/44	
------------	----	---	----	-------------	--

Designated States (National): AT AU BB BG BR BY CA CH CZ DE DK ES FI GB

HU JP KP KR KZ LK LU MG MN MW NL NO NZ PL PT RO RU SD SE SK UA VN

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL

OA PT SE

AU 9458012	A			G06F-009/44	Based on patent WO 9415283
------------	---	--	--	-------------	----------------------------

US 5452456	A		21	G06F-009/44	
------------	---	--	----	-------------	--

EP 674783	A1	E	39	G06F-009/44	Based on patent WO 9415283
-----------	----	---	----	-------------	----------------------------

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LI LU NL

PT SE

JP 8506195	W		43	G06F-009/455	Based on patent WO 9415283
------------	---	--	----	--------------	----------------------------

EP 674783	B1	E		G06F-009/44	Based on patent WO 9415283
-----------	----	---	--	-------------	----------------------------

Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LI LU NL

PT SE

DE 69321255	E			G06F-009/44	Based on patent EP 674783
-------------	---	--	--	-------------	---------------------------

Based on patent WO 9415283

CA 2152041	C			G06F-009/44	
------------	---	--	--	-------------	--

US 6256658	B1			G06F-009/00	
------------	----	--	--	-------------	--

Cont of application US 92993923

Cont of patent US 5452456

JP 2004046881	A		23	G06F-009/455	Div ex application JP 94515254
---------------	---	--	----	--------------	--------------------------------

JP 3508775	B2		22	G06F-009/455	Previous Publ. patent JP 8506195
------------	----	--	----	--------------	----------------------------------

Based on patent WO 9415283

JP 2005149532	A		24	G06F-009/30	Div ex application JP 2003277282
---------------	---	--	----	-------------	----------------------------------

JP 3705547	B2		21	G06F-009/455	Div ex application JP 94515254
------------	----	--	----	--------------	--------------------------------

Previous Publ. patent JP 2004046881

Abstract (Basic): WO 9415283 A

The system has the ability to switch between execution of a number of **object** code types having different conventions for **invoking program procedures** and performing stack manipulations. It may also be used to switch between different calling conventions within a single **object code** type.

The system has a routine descriptor (20), a stack switch frame, a mode switching mechanism for switching from a first processor, **code** or calling convention to second type and processes for executing instructions in various **code** types. When a routine calls a routine having a different stack model, the mode switching mechanism uses the stack switch **frame** to provide a **transition** between the two different stack types.

ADVANTAGE - System allows **code** of any **code** type to execute any other **code** without knowing **code** type of that **code**.

Dwg.2/7

Title Terms: PROCESSOR; SYSTEM; EXECUTE; **PROGRAM** ; CONTAIN; DIFFER; TYPE; **OBJECT** ; **CODE** ; PROCEDURE; FIELD; DIFFERENTIAL; CALL; CONVENTION; CAN; SWITCH; MODE; APPROPRIATE; **OBJECT** ; **CODE**

Derwent Class: T01

International Patent Class (Main): G06F-009/00; G06F-009/30; G06F-009/44; G06F-009/455

International Patent Class (Additional): G06F-009/40; G06F-009/42

File Segment: EPI

37/5/22 (Item 15 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

008891128 **Image available**

WPI Acc No: 1992-018397/ **199203**

XRPX Acc No: N92-013954

Object orientated program for window based computers - has event drivers in window tools which communicate with application using unmodified language C

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: NAYEEM I; ISLAM N

Number of Countries: 002 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
FR 2661525	A	19911031	FR 915113	A	19910425	199203 B
US 5446902	A	19950829	US 90515427	A	19900427	199540
			US 9391536	A	19930714	

Priority Applications (No Type Date): US 90515427 A 19900427; US 9391536 A 19930714

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
-----------	------	--------	----------	--------------

US 5446902	A	17	G06F-007/00	Cont of application US 90515427
------------	---	----	-------------	---------------------------------

Abstract (Basic): FR 2661525 A

A window based computer system can be provided with an **object** orientated **program** using language C. A typical window based system has window tools (30) and library (28) linked to an **application** (32), also interface units and window server (10) (18), the whole being connected to the general system (24).

The window library provides a protocol **procedure** interface and the **application** **calls** the event drivers in the window tools to send **window** and drawing **orders** to the server which notifies events to the **application** in response to the users actions.

ADVANTAGE - **Object** orientated programming is obtained using language C without modification. (35pp Dwg.No.1/7)

Title Terms: **OBJECT** ; ORIENT; **PROGRAM** ; WINDOW; BASED; COMPUTER; EVENT; DRIVE; WINDOW; TOOL; COMMUNICATE; APPLY; UNMODIFIED; LANGUAGE

Derwent Class: T01

International Patent Class (Main): G06F-007/00

International Patent Class (Additional): G06F-009/44

File Segment: EPI

37/5/23 (Item 16 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 Thomson Derwent. All rts. reserv.

008569683 **Image available**

WPI Acc No: 1991-073718/ **199110**

XRPX Acc No: N91-056981

Object based computer system in windows environment - has object

**manager controlling message passing by objects and maintaining
catalogue of objects**

Patent Assignee: HEWLETT-PACKARD CO (HEWP)

Inventor: DUGGAN H; MOREL W

Number of Countries: 013 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 9102307	A	19910221				199110 B
EP 433433	A	19910626	EP 90910820	A	19900731	199126
US 5619638	A	19970408	US 91671806	A	19910503	199720
			US 94203516	A	19940228	

Priority Applications (No Type Date): GB 8917490 A 19890731

Cited Patents: 3.Jnl.Ref; EP 339221

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
WO 9102307	A			
				Designated States (National): US
				Designated States (Regional): AT BE CH DE DK ES FR GB IT LU NL SE
EP 433433	A			
				Designated States (Regional): DE FR GB
US 5619638	A	17	G06F-003/00	Cont of application US 91671806

Abstract (Basic): WO 9102307 A

The system (10) comprises an operating system (12), windows interface (14), **object** manager (16); and **software** for message switching (18), local area networking (20), windows (22) and semantic **objects** (24). The **object** manager controls message passing by **objects** and maintains a catalogue of **objects** residing in the system. It also activates and deactivates **objects** and acts as a library providing primitive utility **functions** to **objects** and other processes.

The distributed message switching **software** is a terminate and stay resident **program** that is used as a message router for the **object** manager. The local area network **software** comprises two terminate and stay resident **programs**.

ADVANTAGE - Provides cross references and updating to changes in data files made by other users. (33pp Dwg.No.6/10)

Title Terms: **OBJECT** ; BASED; COMPUTER; SYSTEM; WINDOW; ENVIRONMENT;
OBJECT ; MANAGE; CONTROL; MESSAGE; PASS; **OBJECT** ; MAINTAIN; CATALOGUE;
OBJECT

File 348:EUROPEAN PATENTS 1978-2006/Feb w03

(c) 2006 European Patent Office

File 349:PCT FULLTEXT 1979-2006/UB=20060223,UT=20060216

(c) 2006 WIPO/Univentio

Set	Items	Description
S1	2702874	PROGRAM? ? OR SOFTWARE OR APPLICATION? ? OR APP OR APPS OR CODE
S2	807500	PANEL? ? OR PANE OR PANES OR WINDOW? ? OR SCREEN? ? OR BOX OR BOXES OR FRAME OR FRAMES
S3	190320	(SERIES OR SEQUENCE? ? OR SUCCESSIV? OR SUCCESSION OR SUCCEEDING OR SUBSEQUENT?? OR TRANSITION? OR ORDER??? OR CONSECUTIV? OR PROGRESSION OR CHAIN??? OR TRAIN? ? OR LINK??? OR CONNECT???) (5N)S2
S4	23692	(NEXT OR ENSUING OR FOLLOWING) (3W)S2
S5	4750	S2(7N)(GRAPH? ? OR GRAPHED OR GRAPHING OR DIGRAPH?)
S6	75295	(INVOK??? OR INVOCATION OR CALL???) (5N)(METHOD? ? OR PROCEDURE? ? OR FUNCTION? ?)
S7	923956	OBJECT? ? OR OO OR OOP OR OOPPL OR OOPLA OR OLE
S8	5663	S1(5N)(LEGACY OR OLD OR OLDER OR OUTDATED OR OUT(1W)DATE OR PREEXIST? OR PRE()EXIST?)
S9	99982	S1(5N)(MODERN OR NEW OR UPDAT? OR CURRENT OR RECENT OR BACKWARD()COMPAT?)
S10	7625	(S1 OR OBJECT? ? OR COMPONENT? ?) (5N)(REUSE? ? OR REUSING - OR REUSABLE OR REUSABILITY OR RE()USE OR USING OR USABLE OR - USABILITY))
S11	328	S1(50N)S3:S5(50N)S6(50N)S7
S12	250	S8:S10 AND S11
S13	80	S12 AND AC=US/PR AND AY=(1978:1995)/PR
S14	80	S12 AND AC=US AND AY=1978:1995
S15	80	S12 AND AC=US AND AY=(1978:1995)/PR
S16	64	S12 AND PY=1978:1995
S17	81	S13:S16
S18	81	IDPAT (sorted in duplicate/non-duplicate order)
S19	78	S11 NOT S12
S20	9	S19 AND AC=US/PR AND AY=(1978:1995)/PR
S21	9	S19 AND AC=US AND AY=1978:1995
S22	9	S19 AND AC=US AND AY=(1978:1995)/PR
S23	14	S19 AND PY=1978:1995
S24	17	S20:S23
S25	682	S1(100N)S3:S5(100N)S6(100N)S7
S26	354	S25 NOT S11
S27	72	S26 AND AC=US/PR AND AY=(1978:1995)/PR
S28	72	S26 AND AC=US AND AY=1978:1995
S29	72	S26 AND AC=US AND AY=(1978:1995)/PR
S30	58	S26 AND PY=1978:1995
S31	81	S27:S30
S32	81	IDPAT (sorted in duplicate/non-duplicate order)

24/3,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2006 European Patent Office. All rts. reserv.

00675866

Information catalog system with object-dependent functionality
Informationsarchivierungssystem mit objektabhängiger Funktionalität
Système d'archivage d'informations avec une fonctionnalité dépendant de l'objet

PATENT ASSIGNEE:

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (Proprietor designated states: all)

INVENTOR:

Harper, Lloyd, 7144 Via Romera, San Jose, California 95139, (US)
Labrie, Jacques, 1415 Hervey Lane, San Jose, California 95125, (US)

LEGAL REPRESENTATIVE:

Burt, Roger James, Dr. (52152), IBM United Kingdom Limited Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)

PATENT (CC, No, Kind, Date): EP 647909 A1 950412 (Basic)
EP 647909 B1 030416

APPLICATION (CC, No, Date): EP 94306033 940816;

PRIORITY (CC, No, Date): US 134355 931008

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS (V7): G06F-017/30

ABSTRACT WORD COUNT: 170

NOTE:

Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB95	383
CLAIMS B	(English)	200316	492
CLAIMS B	(German)	200316	477
CLAIMS B	(French)	200316	647
SPEC A	(English)	EPAB95	5075
SPEC B	(English)	200316	5126
Total word count - document A			5458
Total word count - document B			6742
Total word count - documents A + B			12200

...SPECIFICATION creating an object type. The category subclass, various names and tags, together with the properties of this **object** type may all be specified in an interactive graphical input session. Fig. 8 illustrates a graphical user interface panel 220 for creating an **object** instance. The **object** type may be specified, together with the values of the properties defined by the **object** type, in an interactive graphical input session. Other graphical user interface panels for manipulating **object** types and **object** instances may also generated by the database catalog system 26 to provide functionality similar to that of...

...illustrates an alternative batch input method using a series of tag language statements 230 for defining an **object** type. **Object** instances may be defined in similar fashion. A third alternative input method is to utilize all **application** programming interface (API) allowing access to the database cataloging system via all external **application** using C language **function calls**.

Once the metadata store 28 is generated by creating one or more **object** types and **object** instances in accordance with the methods described above, database cataloging system 26 allows ...within the context of an interactive graphical user interface environment. Referring now to Figs. 10-14, a **series** of exemplary graphical user interface **panels** intended for use by knowledge workers to execute various database cataloging and display functions are shown. Fig...

...SPECIFICATION creating an object type. The category subclass, various names and tags, together with the properties of this **object** type may all be specified in an interactive graphical input session. Fig. 8 illustrates a graphical user interface panel 220 for creating an **object** instance. The **object** type may be specified, together with the values of the properties defined by the **object** type, in an interactive graphical input session. Other graphical user interface panels for manipulating **object** types and **object** instances may also generated by the database catalog system 26 to provide functionality similar to that of tag language statements 230 for defining an **object** type. **Object** instances may be defined in similar fashion. A third alternative input method is to utilize an **application** programming interface (API) allowing access to the database cataloging system via an external **application** using C language **function calls**.

Once the metadata store 28 is generated by creating one or more **object** types and **object** instances in accordance with the methods described above, database cataloging system 26 allows knowledge workers to access ...

...within the context of an interactive graphical user interface environment. Referring now to Figs. 10-14, a **series** of exemplary graphical user interface **panels** intended for use by knowledge workers to execute various database cataloging and display functions are shown. Fig...

24/3,K/7 (Item 7 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2006 European Patent Office. All rts. reserv.

00572645

System and methods for computer interfaces
System und Verfahren für Rechnerschnittstellen
Système et méthodes pour interfaces ordinateur

PATENT ASSIGNEE:

Borland Software Corporation, (1423877), 1800 Green Hills Road, Scotts Valley, California 95066, (US), (Proprietor designated states: all)

INVENTOR:

Anderson, Charles R., 323 Arroyo Seco, Santa Cruz, California 95060, (US)
Warfield, Robert W., 3470 Merrill Street, Aptos, California 95003, (US)
Cseri, Istvan, 15 Terrace View Court, Scotts Valley, California 95066, (US)

Low, Murray K., 802 Columbia Street, Santa Cruz, California 95060, (US)
Liaw, Weikuo, 104 Nanna Court, Santa Cruz, California 95060, (US)
Bush, Alan M., 905 Fremont Place, No.4, Menlo Park, California 94025, (US)

LEGAL REPRESENTATIVE:

Godsill, John Kenneth et al (31031), Haseltine Lake & Co., Imperial House, 15-19 Kingsway, London WC2B 6UD, (GB)

PATENT (CC, No, Kind, Date): EP 569133 A2 931110 (Basic)
EP 569133 A3 940209
EP 569133 B1 011031

APPLICATION (CC, No, Date): EP 93302717 930407;

PRIORITY (CC, No, Date): US 866658 920408

DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE

INTERNATIONAL PATENT CLASS (V7): G06F-017/60; G06F-009/44; G06F-003/033

ABSTRACT WORD COUNT: 93

NOTE:

Figure number on first page: 2A

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	1063

CLAIMS B	(English)	200144	1920
CLAIMS B	(German)	200144	1846
CLAIMS B	(French)	200144	2278
SPEC A	(English)	EPABF1	11496
SPEC B	(English)	200144	11589
Total word count - document A			12560
Total word count - document B			17633
Total word count - documents A + B			30193

...SPECIFICATION inspection (and setting) of properties for graphs is illustrated. Graph window 700 includes a plurality of graph **objects**, each of which may be customized through use of a corresponding property inspector of the present embodiment...

...the part (object) of the graph he or she wishes to change. A right-click on the **graph window** object 710, for example, will invoke the **graph window** inspector 715; at this point, the user may inspect and set various properties of the **graph window** object. In a similar manner, other **objects** of the **graph window** may be inspected. For example, inspection of **graph** background 720 invokes inspector 725, Y-axis **object** 730 invokes inspector 735, X-axis 740 invokes inspector 745, area fill **object** 750 invokes inspector 755, bar series **object** 760 invokes inspector 765, and bar series **object** 770 invokes inspector 775.

Internal Operations

A. Introduction

Internal operations of the system 100 will now be described in detail. In general, operation is driven by **methods** which are invoked by Windows' message dispatcher in response to system or user events. The general mechanism...

...C., Programming Windows, Second Edition, Microsoft Press, 1990. Additional information can be found in Microsoft's Window **Software** Development Kit, including: 1) Guide to Programming, 2) Reference, Vols. 1 and 2, and 3) Tools, all...

...SPECIFICATION inspection (and setting) of properties for graphs is illustrated. Graph window 700 includes a plurality of graph **objects**, each of which may be customized through use of a corresponding property inspector of the present embodiment...

...the part (object) of the graph he or she wishes to change. A right-click on the **graph window** object 710, for example, will invoke the **graph window** inspector 715; at this point, the user may inspect and set various properties of the **graph window** object. In a similar manner, other **objects** of the **graph window** may be inspected. For example, inspection of **graph** background 720 invokes inspector 725, Y-axis **object** 730 invokes inspector 735, X-axis 740 invokes inspector 745, area fill **object** 750 invokes inspector 755, bar series **object** 760 invokes inspector 765, and bar series **object** 770 invokes inspector 775.

Internal Operations

A. Introduction

Internal operations of the system 100 will now be described in detail. In general, operation is driven by **methods** which are invoked by Windows' message dispatcher in response to system or user events. The general mechanism...

...C., Programming Windows, Second Edition, Microsoft Press, 1990. Additional information can be found in Microsoft's Window **Software** Development Kit, including: 1) Guide to Programming, 2) Reference, Vols. 1 and 2, and 3) Tools, all...

(c) 2006 European Patent Office. All rts. reserv.

00571228

Computer system with graphical user interface for window management.
Rechnersystem mit Benutzerschnittstelle für Fensterverwaltung.
Système d'ordinateur avec interface utilisateur graphique pour gestion de
fenêtres.

PATENT ASSIGNEE:

International Business Machines Corporation, (200120), Old Orchard Road,
Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB)

INVENTOR:

Mitsui, Kinichi, 103 Elite, 1-7-5 Kitahara, Asaka-shi, Saitama-ken, (JP)
Javey, Shahram, 377 Elm Road, Toronto, Ontario M5M 3V7, (CA)

LEGAL REPRESENTATIVE:

Burt, Roger James, Dr. (52152), IBM United Kingdom Limited Intellectual
Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)

PATENT (CC, No, Kind, Date): EP 558224 A1 930901 (Basic)

APPLICATION (CC, No, Date): EP 93301112 930216;

PRIORITY (CC, No, Date): JP 9243276 920228

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS (V7): G06F-003/033;

ABSTRACT WORD COUNT: 166

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	398
SPEC A	(English)	EPABF1	4307
Total word count - document A			4705
Total word count - document B			0
Total word count - documents A + B			4705

...SPECIFICATION dividing the window into four areas arranged up-down and left-right. The type of each sub- **window** can be selected from list, **graph** , text, etc. The list arranges data such as names in one direction, and is suitable for displaying a list of information. In terms of the **program** browser, it is, for example, a list of function names or a list of variable names. The...

...a two-dimensional or three-dimensional space by lines, and is suitable for displaying relational information between **objects** . In terms of the **program** browser, it is a representation of a **calling** relationship between **functions** or the like. The text represents characters in a two-dimensional plane, and is suitable for displaying characters themselves of a source file of a **program** or the like. The text sub-window is used as a text editor. If the type of...

24/3,K/9 (Item 9 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

(c) 2006 European Patent Office. All rts. reserv.

00504097

OBJECT BASED COMPUTER SYSTEM
OBJEKTBASIERTES RECHNERSYSTEM
SYSTEME INFORMATIQUE ORIENTE OBJETS

PATENT ASSIGNEE:

Hewlett-Packard Company, (206030), 3000 Hanover Street, Palo Alto,
California 94304, (US), (applicant designated states: DE;FR;GB)

INVENTOR:

DUGGAN, Hugh Hewlett-Packard Limited Filton Road, Stoke Gifford, Bristol
BS12 6QZ, (GB)

MOREL, William, Paul Hewlett-Packard Limited, Filton Road Stoke Gifford,
Bristol BS12 6QZ, (GB)

Robson, Christopher Hewlett-Packard Ltd., Felton Road, Stoke Gifford
Bristol BS12 6QZ, (GB)

LEGAL REPRESENTATIVE:

Squibbs, Robert Francis et al (36277), Intellectual Property Section
Building 2 Hewlett-Packard Limited Filton Road, Stoke Gifford Bristol
BS12 6QZ, (GB)

PATENT (CC, No, Kind, Date): EP 527833 A1 930224 (Basic)
EP 527833 B1 980603
WO 9117499 911114

APPLICATION (CC, No, Date): EP 91908881 910430; WO 91GB690 910430

PRIORITY (CC, No, Date): GB 9009699 900430

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS (V7): G06F-009/44;

NOTE:

No A-document published by EPO

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9823	413
CLAIMS B	(German)	9823	406
CLAIMS B	(French)	9823	472
SPEC B	(English)	9823	3587
Total word count - document A			0
Total word count - document B			4878
Total word count - documents A + B			4878

...SPECIFICATION ERASE"; "INSERT" and "SPECIAL" buttons 30, 31 and 32 whereby modifications may be made to the sequence **object** .

Referring to Figure 7 there is ...viewer sends a message to the sequence object. This is received at 41 and invokes the AddViewer **procedure** 42. The sequence object thereby registers the viewer and establishes a link with the viewer by the...

...described above. Additionally, however, another viewer is added to the sequence object, namely a control viewer, the **procedure** 44 "AddControl Viewer" being **invoked** automatically. In this way the central panel shown in Figure 6 is recovered from a stored file...

...procedure (Figure 7) is effective to pass to the sequence object control messages appropriate to the control **panel** .

The **sequence** object includes a path list 45. This is a list of objects identified by unique object identifiers...

...addresses in the system data structure. The list is created by linking a viewer to the sequence **object** and navigating the viewer around the **object** world in a required path, or sequence. If a particular **object** being viewed is to be included in the path then the "insert" button 11 (Figure 6) is clicked by the mouse. If an **object** is to be removed from the sequence then the "erase" button 10 (Figure 6) is clicked when the **object** is being viewed. These actions **invoke** the **procedures** "AddObjId" 46 and "RemoveObjId" 47 respectively.

If the "Play" button 7 (Figure 6) is clicked then the sequence of **objects** in list 45 is followed and viewed at a predetermined speed. A "StartPlaying" **procedure** 48 is **invoked** to this end and this has the effect of making the sequence **object** a virtual viewer which is coupled to the user's viewer. The same navigated coupling messages as...

...CLAIMS system having at least one computer (PC) programmed to support objects with a processor unit (CPU) and **program** and data storage facilities (RAM, MS), the computer having an object manager for managing a plurality of...

...whereby the viewer can follow the predetermined path, while allowing access by the viewer (4) to semantic **objects** along said predetermined path.

2. An **object** based computer system as claimed in Claim 1 wherein there is provided a coupling mechanism (12, 14...

- ...viewers may indicate and/or may follow the navigation positions of the said one viewer, the sequence **object** utilising the same kind of navigational signals and thus behaving as a viewer coupled to the viewer to which it is linked.
3. An **object** based computer system as claimed in either of the preceding claims wherein there is provided a control **panel** representation file and a **linking** procedure effective to link a viewer (25) with a sequence **object** (26) automatically **invokes** a control panel **procedure** which overlays a control panel on the viewer.
 4. An **object** based computer system as claimed in Claim 3 wherein the control panel is mouse operable and has...
- ...FAST FORWARD" and "REWIND" (27, 28, 29) in a manner analogous to a tape player.
5. An **object** based computer system as claimed in Claim 4 wherein the control panel has representations for "ERASE" and...
- ...the path data and insert into the path data information pertaining to the identity of the current **object** being viewed.
6. An **object** based computer system as claimed in any of the preceding claims wherein the path data is in the form of node addresses of **objects** to be viewed.

24/3,K/16 (Item 2 from file: 349)
 DIALOG(R)File 349:PCT FULLTEXT
 (c) 2006 WIPO/Univentio. All rts. reserv.

00200138

**OBJECT BASED COMPUTER SYSTEM
 SYSTEME INFORMATIQUE ORIENTE OBJETS**

Patent Applicant/Assignee:

HEWLETT-PACKARD COMPANY,
 DUGGAN Hugh,
 MORELL William,

Inventor(s):

DUGGAN Hugh,
 MORELL William,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9117499 A1 **19911114**
 Application: WO 91GB690 19910430 (PCT/WO GB9100690)
 Priority Application: GB 909699 19900430

Designated States:

(Protection type is "patent" unless otherwise stated - for applications prior to 2004)

AT BE CH DE DK ES FR GB GR IT JP LU NL SE US

Publication Language: English

Fulltext Word Count: 4197

Patent and Priority Information (Country, Number, Date):

Patent: ... **19911114**

Fulltext Availability:

Detailed Description
 Claims

Publication Year: **1991**

Detailed Description

... ERASE"; "INSERT" and
 "SPECIAL" buttons 30, 31 and 32 whereby modifications
 may be made to the sequence **object** .

Referring to Figure 7 there is shown
 schematically ...sends a message to
 the sequence object. This is received at 41 and

- 13

invokes the **AddViewer procedure** 42. The sequence object thereby registers the viewer and establishes a link with the viewer by the viewer is added to the sequence object, namely a control viewer, the **procedure** 44 "AddControl Viewer" being **invoked** automatically. In this way the central panel shown in Figure 6 is recovered from a stored file...

...procedure (Figure 7) is effective to pass to the sequence object control messages appropriate to the control **panel** .

The **sequence** object includes a path list 45.

This is a list of objects identified by unique object identifiers...

...addresses in the system data structure. The list is created by linking a viewer to the sequence **object** and navigating the viewer around the **object** world in a required path, or sequence. if a particular **object** being viewed is to be included in the path then the "insert" button 11 (Figure 6) is clicked by the mouse. If an **object** is to be removed from the sequence then the "erase" button 10 (Figure 6) is clicked when the **object** is being viewed, These actions **invoke** the **procedures** "AddObjId" 46 and "RemoveObjId" 47 respectively. If the "Play" button 7 (Figure 6) is clicked then the sequence of **objects** in list 45 is followed and viewed at a predetermined speed. A "StartPlaying" **procedure** 48 is **invoked** to this end and this has the effect of making the sequence **object** a virtual viewer which is coupled to the user's viewer. The same navigated coupling messages as...

Claim

... based computer system having at least one computer with a central processor unit, random access memory and **program** and data storage facilities, the computer being programmed to run in a windows environment and having an...

...of being linked to viewers whereby the viewers can follow the predetermined path, while allowing normal viewer/ **object** observation and manipulation.

2 An **object** based computer system as claimed in Claim 1 wherein there is provided a coupling mechanism whereby different...

...other viewers may indicate and/or follow the navigation positions of the said one viewer, the sequence **object** utilising the same kind of navigational signals and thus behaving as a viewer coupled to the viewer to which it is linked.
3 An **object** based computer system as claimed in either of the preceding claims wherein there is provided a control **panel** -representation file and a **linking** procedure effective to link a viewer with a **sequence object** automatically **invokes** a control **panel procedure** which overlays a control panel on the viewer.

4 An **object** based computer system as claimed in Claim 3 wherein the control panel is mouse operable and has...

...which allow
"PLAY"; "FAST FORWARD" and "REWIND" in a manner analogous to a tape player.

5 An **object** based computer system as claimed in Claim 4 wherein the control panel has representations for "ERASE" and...

...the path
data and insert into the path data information pertaining to the identity of the current **object** being viewed.

6 An **object** based computer system as claimed in any of the preceding claims wherein the path data is in the form of node addresses of **objects** to be viewed.